

# UC Santa Barbara

## UC Santa Barbara Electronic Theses and Dissertations

### Title

Optimal Control and Coordination of Small UAVs for Vision-based Target Tracking

### Permalink

<https://escholarship.org/uc/item/0s18p519>

### Author

Quintero, Steven Andrew Provencio

### Publication Date

2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA  
Santa Barbara

# Optimal Control and Coordination of Small UAVs for Vision-based Target Tracking

A dissertation submitted in partial satisfaction  
of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Steven Andrew Provencio Quintero

Committee in Charge:

Professor João P. Hespanha, Chair

Professor Michael Ludkovski

Professor Francesco Bullo

Professor Katie Byl

September 2014

The dissertation of  
Steven Andrew Provencio Quintero is approved:

---

Professor Michael Ludkovski

---

Professor Francesco Bullo

---

Professor Katie Byl

---

Professor João P. Hespanha, Committee Chair

June 2014

Optimal Control and Coordination of Small UAVs for Vision-based Target  
Tracking

Copyright © 2014

by

Steven Andrew Provencio Quintero



To my mom and dad

## Acknowledgements

I would like to begin by thanking my advisor, Professor João P. Hespanha, for his exceptional guidance and patience over the course of my graduate career. I have benefited greatly from your ability to explain concepts, ideas, and technical details in an intuitive and accessible manner. Thank you for being thoughtful and considerate in the applied research projects (especially GeoTrack) that you have had me work on, as I have greatly enjoyed the nature of the work as well as the ability to field test my control algorithms, which is quite a privilege. Lastly, thank you for showing me how to write well and give a good talk. I certainly haven't arrived in either of these areas, but I do believe my skills in these areas have improved considerably since I've started here at UCSB.

Next I would like to thank Professor Michael Ludkovski both for helping me with my research and serving on my committee. I really appreciate the numerous times you have visited my lab to discuss RMC with me and João to help us get it working for my particular research application. I consider it a privilege to have worked with you and look forward to the publication of our joint work together. Thank you for your patience in helping make the research a success. And thank you for a thorough review of the paper that constitutes the fourth chapter of this dissertation, as I believe it has allowed me to refine the presentation of the material considerably.

I would also like to thank my committee members Professor Francesco Bullo and Professor Katie Byl for serving on my committee. I appreciate your insightful questions and comments and am still considering some of them as future research topics. I have also enjoyed and benefited from the courses I took with each of you, namely Distributed Control of Robotic Networks and Kalman & Adaptive Filtering. I must point out that the latter course inspired the development of my stochastic target motion model presented in Chapter 2.

It has been a privilege to attend the weekly seminars hosted by the Center for Control, Dynamical-Systems, and Computation (CCDC), as the seminars have exposed me to the work of world-class researchers and even allowed me to meet with some of them. And so I would like to thank the professors who have volunteered their time to serve the academic community as directors of this organization, namely Mustafa Khammash, Andy Teel, Bassam Bamieh, Francesco Bullo, and my advisor João Hespanha. I would also like to thank the helpful administrative assistants of the CCDC as well, namely Anna Lin and Dimple Bhatt.

A special thanks is in order for Val de Veyra, who has answered numerous questions and helped me with various tasks over the years starting from day one until now. You go above and beyond the call of duty, and I wish more staff on campus were like you in that regard. Thank you for your patience and tremendous help, as well as the fun conversations that we've had over the years.

Next I would like to thank my lab mates who have been instrumental in helping me to succeed and develop as a PhD student. First I would like to thank Daniel J. Klein for his help in getting me started with both graduate-level research and writing. I have benefited tremendously from our collaboration and am extremely grateful for your help with my first conference paper. Thank you for writing the software interface to Virtual Cockpit, which was vital for field testing my control algorithms at Camp Roberts and was even used in the successful final demo. You were an exceptionally helpful postdoc to me and a number of others in the lab. I would also like to thank Jason Isaacs for his help and advice as the senior-most person in the lab. Thank you for allowing me to share the latest developments in my research and for many hours of discussion, suggestions, and advice. You too have been an exceptionally helpful postdoc. I've enjoyed our friendship and countless visits to delicious lunch venues. Also, thank you for serving as the coffee supplier for the lab. I would like to thank Josh for his help in my research, especially in helping me write a fast regression-based prediction algorithm for my research on regression Monte Carlo, as it inherently possessed the need for fast software. I would also like to thank you (and Diana) for working on my car several times, which you graciously did even after you graduated.

I would like to thank my newer lab mates as well. First I would like to thank Justin Pearson for reinstalling in me the genuine eagerness and enthusiasm to learn, as well as for helping me with subversion and Linux. Hopefully I will get a chance to benefit from your Mathematica expertise before I completely leave UCSB. Also, thanks for sharing your books with the lab. I would like to thank David Copp for helping me and Jason get started with our latest research project. I look forward to using your robust MPC algorithms someday. I would like to thank the rest of my lab mates for their feedback on dry runs and fun and enlightening discussions as well. Namely, I would like to acknowledge Shaunak, Alexandre, Duarte, Soheil, Farshad, Hari, Rodolfo, Kyriakos, Michelle, and Henrique.

I am also grateful to Toyon Research Corporation for their help in making the GeoTrack project a success. Namely, I would like to thank Gaemus Collins, Mike Wiatt, Chris Stankevitz, and former Toyon employee Paul Filitchkin for making flight testing at Camp Roberts a smooth and streamlined process. Thank you for working with me to get my software and algorithms working properly.

I would also like to acknowledge the financial support of the Institute for Collaborative Biotechnologies, which was provided through grant W911NF-09-D-0001 from the U.S. Army Research Office. I am also grateful to the ECE department for supporting me during the 2007-2008 academic year with the

Distinguished Graduate Research Fellowship and during the present quarter with the Spring 2014 Dissertation Fellowship.

I believe those who inspired, encouraged, and helped me get into grad school are very much deserving of thanks as well. First I would like to thank the McNair Scholars program for providing me with much of the necessary guidance, information, and support necessary to successfully pursue a career in graduate school. I would like to thank Mr. David Brandstein for encouraging me to apply to the program and the late Dr. David Viger for being a tremendous help in helping me write my statement of purpose. An extra special thanks is in order for the late Dr. Gary Gear, who was a stellar undergraduate mentor. Thank you for your constant enthusiasm, encouragement, support, and advice. And thank you for letting me stay in your trailer after the fall semester had ended so I could finish writing my statement of purpose. I would also like to thank you for helping provide the opportunity to do two summer internships at NASA's Dryden Flight Research Center. And last but not least, thank you for encouraging me to look for grad schools in California so I could be near my family. I would also like to thank Olga Diaz for her hard work in helping keep the McNair scholars program a success after the two Davids retired. Lastly, I would also like to thank Dr. Chuck Cone for inspiring and encouraging me to pursue a graduate career in the area of automatic control.

There are a number of people who have provided tremendous spiritual support. I would like to thank Pastor Larry Reichardt for his prayers and exemplary teaching and for being a model of faithfulness, integrity, and steadfastness unto the Lord over the decades I have known him. I would like to thank the pastors and members of West Coast Believers Church for being a strong spiritual support team here in Santa Barbara. I am grateful to Pastors RayGene and Beth Wilson for the love and joy that they brought into my life, and for further teaching, exemplifying, and imparting to me principles of faith, love, and joy. Pastor Lisa, I always enjoyed and looked forward to our time of fellowship while greeting people and thank you for your love and prayers. I have enjoyed the friendship of numerous others at WCBC, and would like to thank these people for their love and prayers: Pastors David and Carol Breed, Roy and Jill Coggeshall, Tom and Terri Melody, Domanique Carlton, Estelle Johansen, Lynn Alonso and her late husband Tim, Cory and Michael Abele, Joe and Sarah Zaragosa, Darlene Santiago, and Susan Reynolds.

I must give a very special thanks to Connie Carpenter, who leads an extraordinary lifestyle of integrity, honor, and devotion to the Lord. You have been a stellar role model and spiritual guide during my time here at UCSB, and I hope to continue learning from your exemplary lifestyle of genuine Christianity. Thank you for your love and prayers.

A very special thanks is also in order for the Kimseys, who are my family's best friends. David, Sofia, and Little David, you have proven to be wonderful, genuine friends over the years, both to my parents and to me. I always enjoy our fellowship, which is second to none, and I would like to thank you for your encouragement, love, and prayers, which helped make this accomplishment possible.

Next I would like to thank my friends and family. I would like to thank my most faithful friend Helen Bui, who has kept in touch and visited me here in Santa Barbara over the years. You too understand the rigors of graduate school, and I've truly enjoyed our friendship, which has stood the test of time. I'm proud of your accomplishments as well. Also, I would like to thank my cousin Monique who recently went to heaven. You have always been a very loving and thoughtful person, and I thank you for all your prayers and birthday cards.

I would like to thank my landlords In Santa Barbara as well, as they not provided me with cozy and peaceful living environments so I could eat, study, and rest, but also proved to be good friends as well. To Tim and Michelle Lee, thank you for feeding me dinner numerous times and for being so considerate and respectful. I've enjoyed the time spent watching movies and tennis matches with you two. To Tom and Ilene Dietrich, thank you for also being considerate



and respectful. You have a wonderful family, and I always enjoy when your grandkids visit.

I would also like to thank all of my grandparents. First, I would like to thank my grandparents Benjamin and Connie Provencio, a.k.a. Grandpa and Grandma, who have been some of the biggest supporters of my education. Thank you for your constant enthusiasm and support, especially the financial support. Grandpa, thank you for hosting delicious barbecues with tri-tip and salsa, and thank you Grandma for always buying me whatever I have needed throughout the years, which you did at unbelievable prices. I would also like to my grandparents Joe and Florence Quintero, a.k.a. Poppy and Nana, for their love and spiritual support. Thank you for your prayers as well, which I know have really made a difference. Poppy, thank you for encouraging me to learn as much as I can, and Nana thank you for cooking for me whenever I come and visit.

I had to save the most important thanksgiving for last, which is owed to my parents. They are not only my biggest helpers, supporters, and encouragers, but they are also my best friends. To my mom Linda, thank you for helping me with all of my errands, cooking food whenever I ask, helping me find the right places to live in Santa Barbara, and for the many other countless things you have done behind the scenes to make this achievement possible. Ann in the UCSB bookstore said you are such a good mother, and I couldn't agree with her more.

Thank you for going all out to help me celebrate this momentous occasion of my graduation. To my dad Cornell, thank you for being such a good role model of hard work, and for your spiritual support in the form of prayer and Biblical advice. Our fellowship is always very stirring, joyous, and refreshing.

In 2 Corinthians 3:5, it is written: “Not that we are sufficient of ourselves to think any thing as of ourselves; but our sufficiency is of God.” This is the truth, as this academic achievement would not have been possible without the grace, mercy, and lovingkindness of my Lord and Savior Jesus Christ. Thank you Lord for the things you have done for me and for the wonderful people you have brought into my life. Evangelist Mario Murillo once said, “The presence and power of God will always open the door for us wherever we need to go.” This accomplishment is a testimony to this truth.

# Curriculum Vitæ

Steven Andrew Provencio Quintero

## Education

June 2009	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara
May 2007	B.S. in Electrical Engineering, Embry-Riddle Aeronautical University

## Experience

2009 - 2014	Graduate Student Researcher, University of California, Santa Barbara
2008 - 2008	Research Intern, UCSB in collaboration with Toyon Research Corporation
2007 - 2007	Undergraduate Student Researcher, NASA Dryden Flight Research Center, Edwards, CA
2006 - 2006	Undergraduate Student Researcher, NASA Dryden Flight Research Center, Edwards, CA

## Publications

S. A. P. Quintero, M. Ludkovski, J. P. Hespanha. “Stochastic Optimal Coordination of Small UAVs for Target Tracking using Regression-based Dynamic Programming,” Note: In preparation.

S. A. P. Quintero and J. P. Hespanha. “Vision-based Target Tracking with a Small UAV: Optimization-based Control Strategies,” Note: In review.

S. A. P. Quintero, G. E. Collins, and J. P. Hespanha. “Flocking with Fixed-Wing UAVs for Distributed Sensing: A Stochastic Optimal Control Approach.” *American Control Conference*, Washington, D.C., June 2013.

S. A. P. Quintero, F. Papi, D. J. Klein, L. Chisci, and J. P. Hespanha, “Optimal UAV Coordination for Target Tracking using Dynamic Programming,” in *Proceedings of the IEEE Conference on Decision and Control*, Atlanta, GA, December 2010.

## Abstract

# Optimal Control and Coordination of Small UAVs for Vision-based Target Tracking

by

Steven Andrew Provencio Quintero

Small unmanned aerial vehicles (UAVs) are relatively inexpensive mobile sensing platforms capable of reliably and autonomously performing numerous tasks, including mapping, search and rescue, surveillance and tracking, and real-time monitoring. The general problem of interest that we address is that of using small, fixed-wing UAVs to perform vision-based target tracking, which entails that one or more camera-equipped UAVs is responsible for autonomously tracking a moving ground target. In the single-UAV setting, the underactuated UAV must maintain proximity and visibility of an unpredictable ground target while having a limited sensing region. We provide solutions from two different vantage points. The first regards the problem as a two-player zero-sum game and the second as a stochastic optimal control problem. The resulting control policies have been successfully field-tested, thereby verifying the efficacy of both approaches while highlighting the advantages of one approach over the other.

When employing two UAVs, one can fuse vision-based measurements to improve the estimate of the target’s position. Accordingly, the second part of this dissertation involves determining the optimal control policy for two UAVs to gather the best joint vision-based measurements of a moving ground target, which is first done in a simplified deterministic setting. The results in this setting show that the key optimal control strategy is the coordination of the UAVs’ distances to the target and not of the viewing angles as is traditionally assumed, thereby showing the advantage of solving the optimal control problem over using heuristics. To generate a control policy robust to real-world conditions, we formulate the same control objective using higher order stochastic kinematic models. Since grid-based solutions are infeasible for a stochastic optimal control problem of this dimension, we employ a simulation-based dynamic programming technique that relies on regression to form the optimal policy maps, thereby demonstrating an effective solution to a multi-vehicle coordination problem that until recently seemed intractable on account of its dimension. The results show that distance coordination is again the key optimal control strategy and that the policy offers considerable advantages over uncoordinated optimal policies, namely reduced variability in the cost and a reduction in the severity and frequency of high-cost events.

# Contents

<b>Curriculum Vitæ</b>	<b>xiv</b>
<b>List of Figures</b>	<b>xx</b>
<b>List of Tables</b>	<b>xxii</b>
<b>List of Algorithms</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Vision-based Target Tracking . . . . .	7
1.2 Organization and Contributions . . . . .	9
<b>2 Optimal Control of a Small UAV for Vision-based Target Tracking</b>	<b>13</b>
2.1 Overview . . . . .	14
2.1.1 Related Work . . . . .	16
2.1.2 Chapter Outline . . . . .	21
2.2 Game Theoretic Control Design . . . . .	21
2.2.1 Game Dynamics . . . . .	22
2.2.2 Cost Objective . . . . .	30
2.2.3 Game Setup and Objective . . . . .	33
2.2.4 Dynamic Programming Solution . . . . .	36
2.3 Stochastic Optimal Control Design . . . . .	40
2.3.1 Overview of Stochastic Dynamics . . . . .	43
2.3.2 Stochastic UAV Kinematics . . . . .	44
2.3.3 Stochastic Target Kinematics . . . . .	48
2.3.4 Control Objective and Dynamic Programming Solution . . . . .	51
2.4 Hardware Setup and Flight Test Results . . . . .	56
2.4.1 Experimental Setup . . . . .	57
2.4.2 Game Theory Results . . . . .	62
2.4.3 Stochastic Optimal Control Results . . . . .	65

2.4.4	Quantitative Comparison . . . . .	68
2.4.5	Wind Considerations . . . . .	71
2.5	Conclusion . . . . .	74
<b>3</b>	<b>Optimal UAV Coordination for Vision-based Target Tracking</b>	<b>76</b>
3.1	Introduction . . . . .	77
3.1.1	Related Work . . . . .	79
3.1.2	Organization of Chapter . . . . .	84
3.2	Problem Formulation . . . . .	85
3.2.1	Vehicle Modeling . . . . .	85
3.2.2	Geolocation Error Covariance . . . . .	87
3.2.3	Problem Statement and Solution . . . . .	98
3.3	Simulation Results . . . . .	100
3.3.1	Study of Simulation Results . . . . .	101
3.3.2	Comparison with Standoff Tracking . . . . .	113
3.4	Conclusion . . . . .	117
<b>4</b>	<b>Stochastic Optimal UAV Coordination for Target Tracking</b>	<b>119</b>
4.1	Problem Formulation . . . . .	122
4.1.1	Stochastic Vehicle Dynamics . . . . .	123
4.1.2	Target-Centric State Space . . . . .	126
4.1.3	Stochastic Optimal Control Objective . . . . .	128
4.2	Regression Monte Carlo . . . . .	129
4.2.1	Standard Technique . . . . .	130
4.2.2	Modified Technique for Policy Generation . . . . .	134
4.2.3	Regression . . . . .	137
4.3	Regression Monte Carlo for Target Tracking . . . . .	142
4.3.1	Modified Algorithm . . . . .	142
4.3.2	Initial Condition Set . . . . .	144
4.3.3	Barrier Function . . . . .	145
4.4	Results . . . . .	147
4.4.1	Problem Setup and Solution Parameters . . . . .	148
4.4.2	RMC Performance . . . . .	152
4.4.3	Nature of Optimal Solution . . . . .	167
4.5	Conclusion . . . . .	169
<b>5</b>	<b>Summary and Future Work</b>	<b>175</b>
5.1	Summary . . . . .	175
5.2	Future Work . . . . .	179

Bibliography	182
A Exploiting Symmetry for Computational Savings in RMC	190



# List of Figures

2.1	Setpoint control of roll angle . . . . .	26
2.2	Instantaneous field of view and horizontal field of regard . . . . .	31
2.3	Vertical field of regard . . . . .	32
2.4	Azimuth cost function . . . . .	34
2.5	Control surface for the game theoretic control policy . . . . .	42
2.6	Monte Carlo simulations to sample roll trajectories . . . . .	45
2.7	Error trajectories over a 2-second ZOH period . . . . .	46
2.8	Sample trajectories of the UAV's stochastic kinematics . . . . .	47
2.9	Standard deviation of target's normally distributed turn rate . . . . .	50
2.10	Sample trajectories of the target's stochastic kinematics . . . . .	52
2.11	Control surface for the stochastic optimal control policy . . . . .	57
2.12	Unicorn/Zagi flying wing . . . . .	59
2.13	Image of a single UAV tracking a ground target at Camp Roberts . . . . .	61
2.14	UAV trajectory with game theoretic control policy . . . . .	62
2.15	Viewing geometry performance with game theory . . . . .	64
2.16	Roll command sequence under the game theoretic control policy . . . . .	65
2.17	UAV trajectory with stochastic optimal control . . . . .	66
2.18	Viewing geometry performance with stochastic optimal control . . . . .	68
2.19	Roll command sequence under the stochastic optimal control policy . . . . .	69
3.1	Propagation of attitude error . . . . .	88
3.2	Growth of GEC for Single UAV . . . . .	94
3.3	Individual and fused geolocation error covariances . . . . .	96
3.4	Fused GEC with one UAV fixed at $(x, y, z) = (90, 0, 40)$ . . . . .	97
3.5	Optimally coordinated UAV trajectories with $v = 4.5$ m/s . . . . .	102

3.6	Optimal performance with $v = 4.5$ m/s . . . . .	103
3.7	Optimally coordinated UAV trajectories with $v = 4.5$ m/s and the baseline altitudes doubled . . . . .	104
3.8	Optimal performance with $v = 4.5$ m/s and the baseline altitudes doubled . . . . .	105
3.9	Fused GEC as a function of separation angle and minimum distance	107
3.10	Fused GEC as a function of separation angle and minimum dis- tance with baseline altitudes doubled . . . . .	108
3.11	Optimally coordinated UAV trajectories with $v = 5$ m/s . . . . .	109
3.12	Optimal performance when $v = 5$ m/s . . . . .	110
3.13	Optimally coordinated UAV trajectories with $v = 10.5$ m/s . . . . .	111
3.14	Optimal performance when $v = 10.5$ m/s . . . . .	112
3.15	UAV trajectories with splay-state controller and $v = 4.5$ m/s . . . . .	115
3.16	Tracking performance with splay-state controller and $v = 4.5$ m/s . . . . .	116
4.1	Sample trajectories of refined stochastic kinematic target model	127
4.2	Partitioning scheme for regression . . . . .	139
4.3	Fused GEC with one UAV fixed at $(x, y, z) = (0, 0, 40)$ . . . . .	146
4.4	Sample trajectories with stochastic optimal control policy . . . . .	152
4.5	Sample performance with stochastic optimal control policy . . . . .	153
4.6	Sample trajectories of uncoordinated control policies . . . . .	156
4.7	Sample performance of uncoordinated control policies . . . . .	157
4.8	Transient response of mean value . . . . .	160
4.9	Transient response of 98 <sup>th</sup> percentile . . . . .	161
4.10	Mean value of stage costs over many 10-minute Monte Carlo sim- ulations . . . . .	162
4.11	Standard deviation of stage costs over many 10-minute Monte Carlo simulations . . . . .	163
4.12	Histogram of steady-state costs . . . . .	164
4.13	Histogram of separation angle with stochastic optimal control policy	168
4.14	Joint probability density of UAV distances for optimal control policy . . . . .	170
4.15	Joint probability density of UAV distances for uncoordinated poli- cies . . . . .	171

# List of Tables

2.1	Optimization Parameters . . . . .	40
2.2	Parameters in UAV and Target Dynamics . . . . .	41
2.3	Sets for State Space Discretization . . . . .	41
2.4	Stochastic target motion parameters . . . . .	50
2.5	Parameters in Stochastic UAV dynamics . . . . .	56
2.6	Sets for State Space Discretization . . . . .	56
2.7	Statistics over 15 minute window . . . . .	69
2.8	Statistics for game theoretic policy in steady wind over 15 minute window . . . . .	74
2.9	Statistics for stochastic optimal control policy in steady wind over 15 minute window . . . . .	74
3.1	Simulation Parameters . . . . .	101
4.1	Stochastic target motion parameters . . . . .	126
4.2	General Parameters . . . . .	148
4.3	RMC Parameters . . . . .	148
4.4	State Space Discretization in One-UAV Scenario . . . . .	155

# List of Algorithms

4.1	Standard Regression Monte Carlo . . . . .	135
4.2	Generating samples of the pathwise cost . . . . .	135
4.3	Regression Monte Carlo for Target Tracking . . . . .	144

# Chapter 1

## Introduction

Small autonomous agents are relatively inexpensive mobile robots capable of reliably performing numerous tasks without any dependency on a human operator. Such tasks include exploration and mapping, search and rescue, surveillance and tracking, and real-time monitoring, to name a few. Moreover, an *autonomous agent* is a ground, aquatic, or aerial robot used to perform a task that requires a significant amount of information gathering, data processing, and decision making without explicit human interaction.

One common practice is to use autonomous agents as sensing platforms to gather the best (most accurate) measurements of a moving object or a dynamic environment. In the former scenario, an unmanned aerial vehicle (UAV) equipped with a gimbaled video camera might be tasked with tracking a randomly moving ground target [41] while in the latter scenario a UAV might be tasked with monitoring severe local storms using mobile Doppler radar [13].

In either case, the autonomous agent must make optimal decisions concerning its motions to minimize measurement uncertainty while being robust to process noise. Such process noise may not only be random, but also strategically adverse, as it can arise from either the moving object of interest, the dynamic environment, any unmodeled dynamics, or some combination of the preceding sources. It is precisely this problem of optimal decision making for measurement gathering with robustness to dynamical uncertainty that is the primary theme of this work. This decision making takes place at the guidance-layer of the autonomous agent, which entails that the control inputs under consideration affect an agent's kinematics. Hence, there is an implicit assumption that autonomous agents have an autopilot (or onboard guidance computer) running low-level feedback loops that regulate motor speeds and control surface deflections to achieve the desired guidance commands, thus allowing the control designer to focus on agent kinematics. Moreover, we focus on generating optimal feedback laws that determine the setpoints for the autopilot's low-level control loops, which directly govern an agent's kinematics. Thus, the guidance controller onboard an autonomous agent will determine setpoints such as wheel speed for wheeled mobile robots (WMRs), turn rate or turn acceleration for an autonomous underwater vehicle (AUV), or bank and pitch angles of a UAV.

A common practice in designing controllers for autonomous agents is that simplifications are often made concerning the kinematic models of the platforms that can pose a hindrance to a real-world implementation. As an illustration, AUVs and UAVs are often modeled as constant-speed planar kinematic unicycles with first order rotational dynamics. More specifically, the control input to such vehicles is the rate of change of the velocity orientation; however, a more appropriate model may instead utilize the angular acceleration of the velocity orientation as the control input [31]. This entails that the kinematic model is fourth order rather than third. While a fourth order model is closer to reality, an autonomous vehicle’s kinematics are most accurately described using a model with six degrees of freedom, which comprise the vehicle’s three-dimensional position and orientation (described using an Euler angle sequence) relative to a fixed external coordinate frame. However, such a model is generally intractable for either analytical or optimization-based control approaches. Thus, to mitigate the effects of both modeling errors and external disturbances, one may incorporate process noise into the reduced-order kinematic model for added robustness. Similarly, one may add disturbance variables and perform a Min-Max optimization. We demonstrate the utility of both approaches in practice.

From an economic standpoint, it is desirable to utilize commercial off-the-shelf (COTS) autonomous vehicles with their existing sensor suite. This gener-

ally entails that the comparatively inexpensive sensors will have limited capabilities. For example, UAVs are often equipped with electro-optical (EO) video cameras, yet these cameras may be in a fixed orientation onboard the aircraft. If a camera is in fact gimballed, it may be that the gimbal mechanism is not capable of continuous pan-tilt motion. In either case, blind spots will exist in the aircraft's visibility region, thereby limiting the trajectories that an agent must make to successfully track or survey an object or region of interest. Hence, an analytical control design becomes more challenging, yet, in a dynamic optimization, one can treat such limitations as soft constraints in the control problem by incorporating the sensor restrictions and limitations into the cost function. This work demonstrates the effectiveness of such an approach in the field.

When using small autonomous agents to gather measurements, one may employ multiple agents to perform a task cooperatively, as such platforms are becoming increasingly common and inexpensive. In this setting, the agents work together to provide synoptic coverage of the desired object or environment of interest, and they can coordinate their behavior to further improve measurement gathering with the existing sensor suite. For example, in hurricane sampling application with  $N$  UAVs, the  $N$  agents may traverse a quadrifolium (a polar rose with 4 petals) to sample each quadrant of the storm [8]. In an ocean sampling application with  $N$  AUVs in a steady underwater current, the agents may stabi-



lize to a circular formation around a point of interest with a constant revisit rate at any point along the circular path in order to provide a diverse set of samples [35]. However, one aspect of multi-agent, and even single-agent applications, is that heuristics are often employed in control designs rather than dynamic optimizations. For example, nonlinear feedback laws or Lyapunov guidance vector fields are often used to stabilize temporal or spatial configurations that should improve some metric of the measurements taken for a particular mission. Of course, tools used to solve dynamic optimizations, e.g., dynamic programming, generally do not scale well with dimensionality of the problem and hence can only address a small number of agents. Nonetheless, a given mission may require only a few agents for satisfactory performance, and consequently, it is worth investigating whether proposed heuristics are truly optimal for a given metric of mission performance. This is the secondary theme of this work, as we show that traditional control strategies for a particular application are quite suboptimal when certain restrictions on agent motion are removed.

Finally, this work focuses on optimally controlling and coordinating autonomous agents modeled as constant-speed nonholonomic vehicles that maintain a fixed altitude or depth. Nonetheless, the approaches taken here are certainly amenable to vehicles that traverse 3-dimensional trajectories, as well as those that have the ability to stop, including quad-rotors, WMRs, and possibly

even flapping-wing Micro Air Vehicles (MAVs). Since the fixed-speed nonholonomic vehicles under consideration have first (or higher) order heading dynamics, associated dynamic optimizations generally require a moderate to long planning horizon for good performance because the benefit of a control action is typically not realized immediately. Moreover, greedy approaches or receding horizon approaches with short planning horizons are often inadequate for satisfactory performance. Dynamic programming is an optimal control methodology that is a powerful tool for solving problems with long planning horizons, and hence it plays a vital role in this work. Moreover, we demonstrate its utility throughout this dissertation, and even demonstrate a simulation-based dynamic programming technique that is able to provide approximate solutions to a 9-dimensional stochastic optimal control problem that only until recently seemed to be intractable on account of its dimension.

We now turn our attention to the particular application of interest, namely *vision-based target tracking* with small, fixed-wing UAVs. This particular application embodies all of the aforementioned themes and design principles while possessing certain properties unique to the particular onboard sensor and autonomous agent platform.

## **1.1 Vision-based Target Tracking**

The task of vision-based target tracking with a single, small UAV entails that an autonomous camera-equipped agent is responsible for gathering the best vision-based measurements of a vehicle moving unpredictably in the ground plane. The qualifier “best” refers to measurements with the least amount of uncertainty, or those whose errors have the smallest covariance. The class of UAVs under consideration in this dissertation are hand or catapult launched fixed-wing aircraft that fly at a constant altitude and are equipped with a global positioning system (GPS), an inertial navigation system (INS), and a gimbaled electro-optical video camera. Additionally, we assume a UAV has an autopilot that regulates airspeed, altitude, and either turn rate or roll angle to the desired setpoints through internal feedback loops. This UAV platform is quite popular due to its well-understood dynamics, comparatively simple design, speed, and endurance. In addition, video cameras are very common sensors that typically come standard in commercially available UAVs due to their light weight, low cost, and ability to provide information about distant objects.

In vision-based target tracking, image processing software determines the centroid pixel coordinates of a ground target moving in the image frame. Given these pixel coordinates, the intrinsic and extrinsic camera parameters, and the

terrain data, one can estimate the three-dimensional location of the target in inertial coordinates and compute the associated error covariance. This is the process of *geolocation* for video cameras [30]. The geolocation error is highly sensitive to the relative position of a UAV with respect to the target. As the UAV's planar distance from the target increases, the associated error covariance grows and becomes significantly elongated in the viewing direction. The smallest geolocation error comes when the UAV is directly above the target, in which case the associated covariance is circular. Thus, a UAV would ideally hover directly above the target, but the relative dynamics between a UAV and target typically preclude this viewing position from being maintained over a period of time. Thus, the control objective becomes having the UAV minimize its distance to the target over time. If, in addition, the UAV has a limited field of regard, or sensing region, then it must maneuver to keep sight of the target. The challenging aspect of this problem is that the underactuated UAV be robust to target maneuvers that are unpredictable and possibly even evasive.

To mitigate the effects of a single UAV's inability to maintain close proximity to the target, one can employ multiple UAVs to gather the best *joint* measurements. In this scenario, the objective is to minimize the *fused* geolocation error covariance of the target position estimate obtained by fusing the individual geolocation measurements. Thus, as in the majority of work on UAV coordination,

we seek optimally coordinated behavior aimed at improving the estimate of the target state. The problem of optimally controlling one or more UAVs to improve target state estimation directly yields a partially observable Markov Decision Process (POMDP), which has an infinite dimensional state space (see §4 of [32]) and is hence avoided.

## 1.2 Organization and Contributions

This dissertation addresses the aforementioned target tracking scenarios and generates robust, practical control policies in each case. Namely, in Chapter 2, we address the scenario in which a single, camera-equipped UAV with a limited field of regard (visibility region) tracks a ground target that moves unpredictably. In this situation, the UAV must maintain close proximity to the ground target to reduce measurement uncertainty and simultaneously keep the target in its camera's field of regard. To achieve this objective robustly, two novel optimization-based control strategies are developed. The first addresses the problem as a two-player zero-sum game with the UAV as the minimizer and the target as the maximizer. The second addresses the problem in the framework of stochastic optimal control, where the target is modeled as a nonholonomic vehicle with stochastic control inputs. Moreover, the first assumes an evasive target motion while the second

assumes a stochastic target motion. In both approaches, dynamic programming is used to generate optimal control policies offline that minimize the expected total cost over a finite horizon, where the individual stage cost is a function of the viewing geometry. The resulting optimal control policies have been successfully flight tested, thereby demonstrating the efficacy of both approaches in a real-world implementation and highlighting the advantages of one approach over the other.

In Chapter 3, we focus on optimally coordinating two UAVs to gather the best *joint* measurements of a moving ground target without any restrictions on sensor visibility. Much work has been proposed for coordinated target tracking without explicitly considering minimization of vision-based geolocation error. Hence, we provide an explicit derivation of the geolocation error covariance using a flat-Earth approximation, following and refining the exposition in [30]. More specifically, we show how errors in the sensor attitude matrix, which relates measurements in the sensor frame to the topographic coordinate frame, amplify errors in the estimate of the target's position in the ground plane. To perform a preliminary analysis of the optimal trajectories free from the effects of stochasticity and higher order dynamics, we model the UAVs as Dubins vehicles and the target as a constant-velocity unicycle and compute the optimal control policies that minimize the fused geolocation error covariance over a long plan-

ning horizon. A surprising result, and the main contribution of this work, is that the dominant factor governing the optimal UAV trajectories is coordination of the distances to the target and not of the viewing directions, as is traditionally assumed.

In Chapter 4, we consider the objective of the previous chapter, but address the problem with a higher degree of realism by using stochastic kinematic models similar to those in Chapter 2. The goal is to remedy the limitations of work that employs heuristics to approximate the results of Chapter 3, as such work is generally non-robust to stochastic target motion and only employs a single trajectory type rather than a mixture of the orbital and sinusoidal trajectories that compose the optimal trajectories. Moreover, the advantage of this approach is that the solution yields a control policy that is robust to real-world phenomenon, readily implemented in the field, and automatically adapts the UAV trajectories to unpredictable target maneuvers. However, this problem formulation yields a 9-dimensional stochastic optimal control problem for which grid-based solutions are infeasible. In order to circumvent this challenge, we present a policy generation technique derived from the simulation-based policy iteration scheme known as regression Monte Carlo, as well as a partitioned robust regression scheme that lies at the heart of the algorithm. We again recover the distance coordination

behavior of the simplified problem formulation and show the advantages of this approach over common alternatives.

In Chapter 5, we summarize our results and contributions and discuss the underlying design principles that we have emphasized and demonstrated throughout this work. We also discuss a number of avenues for interesting future research.



## Chapter 2

# Optimal Control of a Small UAV for Vision-based Target Tracking

In this chapter, we detail the design of two different control policies that enable a small, fixed-wing unmanned aerial vehicle (UAV), equipped with a pan-tilt gimbaled camera, to autonomously track a moving ground vehicle (target). The specific control objective is best described by Saunders in §4.1 of [40], where he defines *vision-based target tracking* as “maintaining a target in the field-of-view of an onboard camera with sufficient resolution for visual detection.”

Specific to the fixed-wing UAV used in the flights experiments is a mechanical limitation of the pan-tilt gimbal mechanism that requires the UAV to keep the target towards its left-hand side for visibility. Nonetheless, by adjusting the cost function of the dynamic optimization, this work can be adapted to the fixed-camera scenario that is common on smaller platforms such as Micro Air Vehicles (MAVs). The sensor visibility constraint coupled with uncertain target motion

and underactuated UAV dynamics compose the control challenge for which two novel solutions are presented.

## 2.1 Overview

Two different styles of optimization-based control policies are developed to enable a small UAV to maintain visibility and proximity to target in spite of sensor blind spots, underactuated dynamics, and evasive or stochastic nonholonomic target motion. The first is a game theoretic approach that assumes evasive target motion. Hence, the problem is formulated as a two-player, zero-sum game with perfect state feedback and simultaneous play. The second is a stochastic optimal control approach that assumes stochastic target motion. Accordingly, in this approach, the problem is treated in the framework of Markov Decision Processes (MDPs). In both problem formulations, the following are key features of the control design:

1. The UAV and the target are modeled by fourth-order discrete-time dynamics, including simplified roll (bank) angle dynamics with the desired roll angle as the control input.
2. The UAV minimizes an expected cumulative cost over a finite horizon.

3. The cost function favors good viewing geometry, i.e., visibility and proximity to the target, with modest pan-tilt gimbal angles.
4. The dynamic optimization is solved offline via dynamic programming.

Both approaches incorporate roll dynamics because the roll dynamics can be on the same time scale as the heading dynamics, even for small (hand-launched) UAVs. Accordingly, this work directly addresses the phase lag in the heading angle introduced by a comparatively slow roll rate that would otherwise be detrimental to the UAV's tracking performance. Additionally, for small aircraft, the range of permissible airspeeds may be very limited, as noted in [7], while frequent changes in airspeed may be either undesirable for fuel economy or unattainable for underactuated aircraft. Thus, both control approaches assume a constant airspeed and treat the desired roll angle of the aircraft as the sole control input that affects the horizontal plant dynamics. The target is modeled as a nonholonomic vehicle that turns and accelerates.

In order to determine control policies (decision rules for the desired roll angle) that facilitate good viewing geometry, a cost function is introduced to penalize extreme pan-tilt angles as well as distance from the target. Dynamic programming is used to compute (offline) the optimal control policies that minimize the expected cumulative cost over a finite planning horizon. The control policies are

effectively lookup tables for any given UAV/target configuration, and hence they are well suited for embedded implementations onboard small UAVs. These control policies have been successfully flight tested on hardware in the field, thereby verifying their robustness to unpredictable target motion, unmodeled system dynamics, and environmental disturbances. Lastly, although steady wind is not directly addressed in the problem formulation, high fidelity simulations were performed that both verify and quantify the policies' inherent robustness to light and even moderate winds.

### **2.1.1 Related Work**

Significant attention has been given to the target tracking problem in the past decade. Research groups have approached this problem from several different vantage points, and hence notable work from these perspectives is now highlighted. One line of research proposes designing periodic reference flight trajectories that enable the UAV to maintain close proximity to the target as it tracks the reference trajectories via waypoint navigation [23] or good helmsman steering [19]. Although one reference trajectory is typically not suitable for all target speeds, one can optimally switch between them based on UAV-to-target speed in order to minimize the maximum deviation from the target [3]. A particularly unique line of work on target tracking is that of oscillatory control of

a fixed-speed UAV. In this approach, one controls the amplitude and phase of a sinusoidal heading-rate input to a kinematic unicycle such that the average velocity along the direction of motion equals that of the ground target, which is assumed to be piecewise constant [22, 38]. None of the preceding works, however, consider any limitations imposed by miniature vision sensors that are common on small, inexpensive UAVs.

Perhaps the greatest amount of research in the general area of target tracking is devoted to solving the specific problem of *standoff tracking*. The control objective for this problem is to have a UAV orbit a moving target at a fixed, planar standoff distance. To achieve this goal, a number of approaches utilize nonlinear feedback control of the UAV's heading rate, wherein vision-sensor requirements are addressed. Dobrokhodov et al. develop control laws for controlling both a UAV and its camera gimbal [11]. The authors design nonlinear control laws to align the gimbal pan angle with the target line-of-sight (LOS) vector and the UAV heading with the vector tangent to the LOS vector; however, only uniform ultimate boundedness is proved. Li et al. advance the previous work by reformulating the control objective, adapting the original guidance law, and proving asymptotic stability of the resultant closed-loop, non-autonomous system [26]. The authors of [26] further adapt this newly designed control law to achieve asymptotic stability for the case of time-varying target velocity, although

it comes at the high cost of requiring airspeed control as well as data that is non-trivially acquired, namely the target's turn rate and acceleration. Saunders and Beard consider using a fixed-camera MAV to perform vision-based target tracking [41]. By devising appropriate nonlinear feedback control laws, they are able to minimize the standoff distance to a constant-velocity target, while simultaneously respecting field of view (FOV) and max roll angle constraints.

Anderson and Milutinović present an innovative approach to the standoff tracking problem by solving the problem using stochastic optimal control [2]. Modeling the target as a Brownian particle (and the UAV as a deterministic Dubins vehicle), the authors employ specialized value iteration techniques to minimize the expected cost of the total squared distance error discounted over an infinite horizon. As no penalty is imposed on the control value, the resulting optimal control policy is a bang-bang turn-rate controller that is highly robust to unpredictable target motion. However, the discontinuous turn rate and absence of sensor limitations render the control policy infeasible in a real-world implementation.

Others have also employed optimal control to address the general target tracking problem, wherein the optimization criterion is mean-squared tracking error. Ponda et al. consider the problem of optimizing trajectories for a single UAV equipped with a bearings-only sensor to estimate and track both fixed and

moving targets [36]. By performing a constrained optimization that minimizes the trace of the Cramer-Rao Lower Bound at each discrete time step, they show that the UAV tends to spiral towards the target in order to increase the angular separation between measurements while simultaneously reducing its distance to the target. While Ponda's approach is myopic, i.e., no lookahead, and controls are based on the true target position, Miller et al. propose a non-myopic solution that selects the control input based on the probability distribution of the target state, where the distribution is updated by a Kalman filter that assumes a nearly constant velocity target model [32]. Moreover, Miller poses the target tracking problem as a partially observable Markov decision process (POMDP) and presents a new approximate solution, as nontrivial POMDP problems are intractable to solve exactly [44].

None of the preceding works have considered a target that performs evasive maneuvers to escape the camera's FOV, yet similar problems have been addressed long ago in the context of differential games [20]. In particular, Dobbie characterized the surveillance-evasion game in which a variable-speed pursuer with limited turn radius strives to keep a constant-speed evader within a specified surveillance region [10]. Lewin and Breakwell extend this work to a similar surveillance-evasion game wherein the evader strives to escape in minimum time, if possible [24]. While the ground target may not be evasive, treating the prob-

lem in this fashion will produce a UAV control policy robust to unpredictable changes in target velocity.

In all of the preceding works, at least one or more assumptions are made that impose practical limitations. Namely, the works mentioned thus far assume at least one of the following:

1. Input dynamics are first order, which implies that roll dynamics have been ignored.
2. Changing airspeed quickly / reliably is both acceptable and attainable
3. Target travels at a constant velocity.
4. Sensor is omnidirectional.
5. Sinusoidal/orbital trajectories are optimal, including those resulting from standoff tracking.

The work presented here removes all of these assumptions to promote a practical, robust solution that can be readily adapted to other similar target tracking applications that may have different dynamics and hardware constraints. The policies also possess an inherent robustness that allow them to even track an unpredictable target in the presence of light to moderate steady winds.



## **2.1.2 Chapter Outline**

The remainder of this chapter is organized as follows. Sections 2.2 and 2.3 detail the game theoretic and stochastic optimal control approaches to vision-based target tracking, respectively. These sections discuss the specific UAV and target dynamical models, the common cost function, and the individual dynamic programming solutions. Section 2.4 describes the experimental hardware setup and also presents the flight test results for each control approach. Furthermore, this section also provides a quantitative comparison of the two approaches and draws conclusions concerning the preferred control approach. Section 2.4 concludes by studying the effects of wind on the performance of the policies in a high fidelity simulation environment to quantify practical upper limits on the wind speeds that can be tolerated. Finally, Section 5 provides conclusions of the overall work and discusses venues for future work.

## **2.2 Game Theoretic Control Design**

This section details the game theoretic approach to vision-based target tracking. The key motivations for this approach are to remedy the usual constant-velocity target assumption seen in much of the literature and also to account for sensor visibility limitations. This is done by assuming the target performs

evasive maneuvers, i.e., it strives to enter the sensor blind spots of the UAV according to some control policy optimized to play against that of the UAV. Accordingly, the problem is posed as a multi-stage, two-player, zero-sum game with simultaneous play and solved with tools from noncooperative game theory. The two main elements of a game are the actions available to the players and their associated cost. Thus, the players' actions at each stage are first described, along with their respective dynamics. The cost function of the viewing geometry is presented next and is the same as that used in the stochastic approach. Lastly, this section presents the formal problem statement and the dynamic programming solution that generates a control policy for each player.

### **2.2.1 Game Dynamics**

While the majority of work on target tracking uses continuous time motion models, this work treats the optimization in discrete time. Thus, each vehicle is initially modeled by fourth-order continuous-time dynamics, and then a  $T_s$ -second zero-order hold (ZOH) is applied to both sets of dynamical equations to arrive at the discrete-time dynamics of the overall system.

The UAV is assumed to have an autopilot that regulates roll angle, airspeed, and altitude to the desired setpoints via internal feedback loops. In the model, the aircraft flies at a fixed airspeed  $s_a$  and at a constant altitude  $h_a$  above the

ground. The UAV's planar position  $(x_a, y_a) \in \mathbb{R}^2$  and heading  $\psi_a \in \mathbb{S}^1$  are measured in a local East-North-Up (ENU) earth coordinate frame while its roll angle  $\phi \in \mathbb{S}^1$  is measured in a local North-East-Down (NED) body frame. In the latter coordinate frame, the  $x$ -axis points out of the nose, the  $y$ -axis points out of the right wing, and the  $z$ -axis completes the right-handed coordinate frame.

Throughout this monograph, the roll/bank angle of the aircraft will be the sole control input that affects the horizontal plant dynamics. Furthermore, we will be controlling the roll angle through setpoint control, which entails that a given control policy will determine the desired roll angle that is provided to the autopilot's low-level control loops. In reality, the roll angle is a continuous quantity; however, for the purpose of computing the optimal control policy, it will be advantageous for us to work with a quantized roll angle  $r$  that is discrete both in time and in value. Thus, at discrete time instances  $t = kT_s$  seconds, where  $k \in \mathbb{Z}_{\geq 0}$ , the discrete (or quantized) roll angle  $r_k = r(kT_s)$  is related to the true roll angle  $\phi_k = \phi(kT_s)$  as follows:

$$r_k := q(\phi_k, C), \quad (2.1)$$

where  $C$  is a finite set of quantization values and, for  $s \in \mathbb{R}^n$  and a set  $X \subset \mathbb{R}^n$ ,

$$q(s, X) := \arg \min_{x \in X} \|s - x\|_1. \quad (2.2)$$

Hence  $q(\phi_k, C)$  maps  $\phi_k$  to the nearest value in  $C$ , and we let

$$C := \{0, \pm\Delta, \pm2\Delta\}, \quad (2.3)$$

where  $\Delta > 0$  is the maximum allowable change in the discrete roll angle  $r$  from one ZOH period to the next. We define the overall UAV state as  $\xi := (x_a, y_a, \psi_a, r)$ .

As noted in [41], typical roll dynamics for a small UAV can be modeled as the following first order system:

$$\frac{d}{d\tau}\phi = -\alpha_\phi(\phi - (r + u)), \quad (2.4)$$

where  $1/\alpha_\phi > 0$  is the time constant corresponding to the autopilot control loop that regulates the actual roll angle  $\phi$  to the *current* roll-angle setpoint  $r + u$ . We shall adopt this model for the purpose of game theoretic control of a small UAV. Also, we shall henceforth denote by  $\bar{r}$  the *current* roll-angle setpoint, or *current* roll command, which is defined as  $\bar{r} := r + u$ . Furthermore, in this framework, we apply a  $T_s$ -second ZOH to both the discrete roll angle  $r$  and the control input  $u$ , which represents the change in  $r$  at the end of the ZOH period. Thus, the solution to this system is

$$\phi(\tau, r, u) := \bar{r}(0) + (\phi(0) - \bar{r}(0)) \exp(-\alpha_\phi \tau), \quad (2.5)$$

where  $\tau \in [0, T_s]$ . This corresponds to the discrete-time system

$$\phi_{k+1} = \bar{r}_k + (\phi_k - \bar{r}_k) \exp(-\alpha_\phi T_s). \quad (2.6)$$

We stipulate that the control input  $u$  belongs to the *roll-angle action space*  $U(r)$ , where

$$U(r) := \{u \in \mathfrak{U} : (r + u) \in C\} \quad (2.7)$$

and

$$\mathfrak{U} := \{0, \pm\Delta\}. \quad (2.8)$$

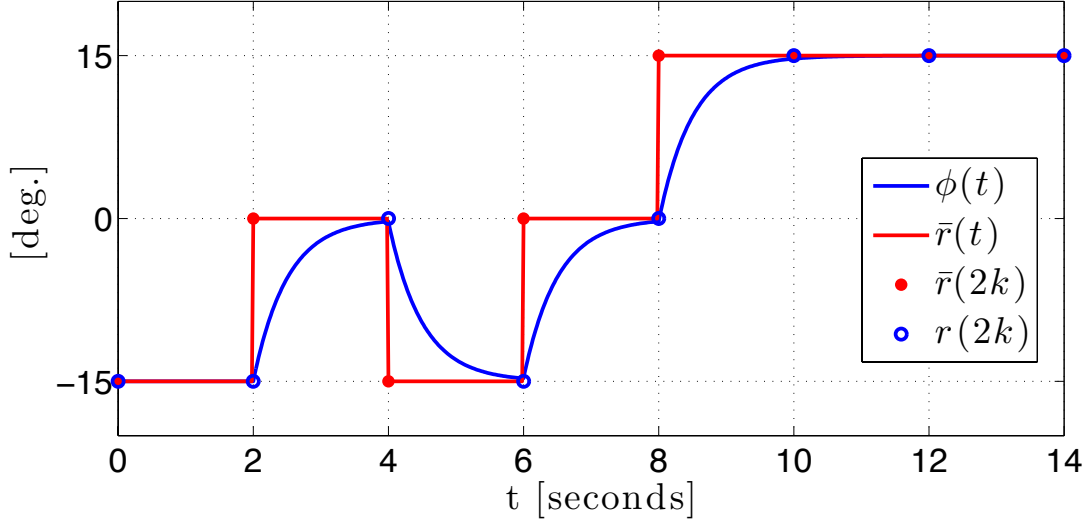
This allows roll commands to change by at most  $\Delta$  and avoids sharp changes in roll that would be detrimental to image processing algorithms in the target tracking task [7].

We note that, for  $\alpha_\phi T_s$  large enough, the roll angle approximately achieves the roll-angle setpoint  $\bar{r}_k = r_k + u_k$  according to (2.6). Moreover,  $\phi_{k+1} \approx r_k + u_k$ , where  $(r_k + u_k) \in C$ , since  $u_k \in U(r_k)$ . Assuming  $|\phi_{k+1} - (r_k + u_k)| < \Delta/2$ , we also have  $q(\phi_{k+1}, C) = q(r_k + u_k, C) = r_k + u_k$ . And furthermore, since  $r_{k+1} := q(\phi_{k+1}, C)$ , we have

$$r_{k+1} = r_k + u_k. \quad (2.9)$$

Hence,  $r$  can be regarded as being natively discrete, both in time and in value.

Figure 2.1 illustrates the previous aforementioned quantities  $\phi$ ,  $r$ , and  $\bar{r}$  and their



**Figure 2.1:** Setpoint control of roll angle with dynamics given by (2.4) with  $\alpha_\phi = 2$  and  $\phi(0) = r(0) = 15^\circ$ . Once every  $T_s = 2$  seconds the discrete roll angle  $r_k$  is changed by  $u_k \in U(r_k)$ , where  $u_k$  is chosen randomly and each element occurs with equal probability. Here, the maximum allowable change in roll angle is  $\Delta = 15^\circ$ .

relationship to one another. The key feature of this plot is that  $r$  approximates  $\phi$  well at the discrete time instances of  $t = kT_s$  seconds.

The UAV's pose (position and heading) dynamics are those of a planar kinematic unicycle. For convenience, we shall define the UAV's pose as  $\mathbf{p} := (x_a, y_a, \psi_a)$ , and the corresponding pose dynamics are

$$\frac{d\mathbf{p}}{d\tau} = \begin{pmatrix} s_a \cos \psi_a \\ s_a \sin \psi_a \\ -(\alpha_g/s) \tan \phi(\tau, r, u) \end{pmatrix}, \quad (2.10)$$

where  $\alpha_g > 0$  is the acceleration due to gravity and  $\phi(\tau, r, u)$  is given by (2.5). Moreover, applying a  $T_s$ -second ZOH to the UAV subsystem produces a discrete-time system  $\xi_{k+1} = f_1(\xi_k, u_k)$ , where  $r_{k+1}$  is given by (2.9) and  $\mathbf{p}_{k+1}$  is the implicit solution to the system of differential equations (2.10) at the end of the ZOH period with initial condition  $\mathbf{p}_k$  and  $\phi(0) = r_k$  in (2.5).

Most work in this area assumes that  $\alpha_\phi$  is large enough so that there is a separation of time scales between the heading dynamics  $\dot{\psi}_a$  in (2.10) and the controlled roll dynamics of (2.4), and consequently, the roll dynamics can be ignored. However, when this assumption does not hold, the resultant phase lag introduced into the system can prove detrimental to target tracking performance. This is the case for small UAVs, like the one used in the experimental work of the present monograph, and hence such a simplifying assumption is avoided. Moreover, the planar kinematic unicycle model used here has second-order rotational dynamics rather than first, the latter of which are encountered more often.

The target is assumed to be a nonholonomic vehicle that travels in the ground plane and has the ability to turn and accelerate. Its state comprises its planar position  $(x_g, y_g) \in \mathbb{R}^2$ , heading  $\psi_g \in \mathbb{S}^1$ , and speed  $v \in \mathbb{R}_{\geq 0}$  and is hence defined as  $\eta := (x_g, y_g, \psi_g, v)$ . The target's dynamics are those of a planar kinematic

unicycle, i.e.,

$$\dot{\eta} = \frac{d}{dt} \begin{pmatrix} x_g \\ y_g \\ \psi_g \\ v \end{pmatrix} = \begin{pmatrix} v \cos \psi_g \\ v \sin \psi_g \\ \omega \\ a \end{pmatrix}, \quad (2.11)$$

where  $\omega$  and  $a$  are the turn-rate and acceleration control inputs, respectively.

Applying a  $T_s$ -second ZOH to the target's control inputs produces a discrete-time system  $\eta_{k+1} = f_2(\eta_k, d_k)$ , which is the solution to (2.11) at the end of the ZOH period with initial condition  $\eta_k$  and  $d_k := (\omega, a)$ . To describe the target's action space, the set of admissible target speeds  $W$  is first introduced, along with the target's maximum speed  $\bar{v}$ , which is assumed to be less than the UAV's airspeed. Denoting the target's maximum acceleration by  $\bar{a}$ , the target's acceleration  $a$  is assumed to belong to the following set:

$$D_a(v) := \begin{cases} \{0, \bar{a}\}, & v = 0 \\ \{-\bar{a}, 0, \bar{a}\}, & v \in W \setminus \{0, \bar{v}\} \\ \{-\bar{a}, 0\}, & v = \bar{v}. \end{cases} \quad (2.12)$$

Furthermore, with  $W = \{0, \bar{a}T, 2\bar{a}T, \dots, \bar{v}\}$ , the set  $W$  is invariant in the sense that  $v \in W$  and  $a \in D_a(v)$  implies  $v^+ \in W$ , as  $v^+ = \bar{a}T + v$ . This property not only enforces speed bounds, but also improves the accuracy of the solution to



the dynamic optimization. Denoting the target's maximum turn rate by  $\bar{\omega}$ , the target's turn-rate  $\omega$  is assumed to take on values in the following set:

$$D_{\omega}(v) := \begin{cases} \{0\} & , v \in \{0, \bar{v}\} \\ \{-\bar{\omega}, 0, \bar{\omega}\} & , v \in W \setminus \{0, \bar{v}\} \end{cases}. \quad (2.13)$$

This restriction on the turn rate implies that the target vehicle cannot turn while stopped nor while traveling at its maximum speed  $\bar{v}$ . Otherwise, it has the ability to turn left, go straight, or turn right using its maximum turn rate. The target's overall action space is defined as

$$D(v) := D_{\omega}(v) \times D_a(v), \quad (2.14)$$

and hence  $d \in D(v)$ . Accordingly, depending on its current speed  $v$ , the target has anywhere from 2 – 9 action pairs from which to choose at a given stage of the game.

The overall 5-dimensional state of the game is denoted by  $z$  and is a combination of the UAV and target states. In particular, the first three components of  $z$  are relative quantities in a target-centric coordinate frame, and the remaining two are absolute. With the relative planar position  $\mathbf{r} \in \mathbb{R}^2$  in the target-centric coordinate frame given by

$$\mathbf{r} = \begin{bmatrix} \cos \psi_g & \sin \psi_g \\ -\sin \psi_g & \cos \psi_g \end{bmatrix} \begin{bmatrix} x_a - x_g \\ y_a - y_g \end{bmatrix}, \quad (2.15)$$

the overall state vector is defined as

$$z := (\mathbf{r}, \psi_a - \psi_g, r, v). \quad (2.16)$$

Hence,  $z$  belongs to the state space  $\mathcal{Z}$ , which is defined as

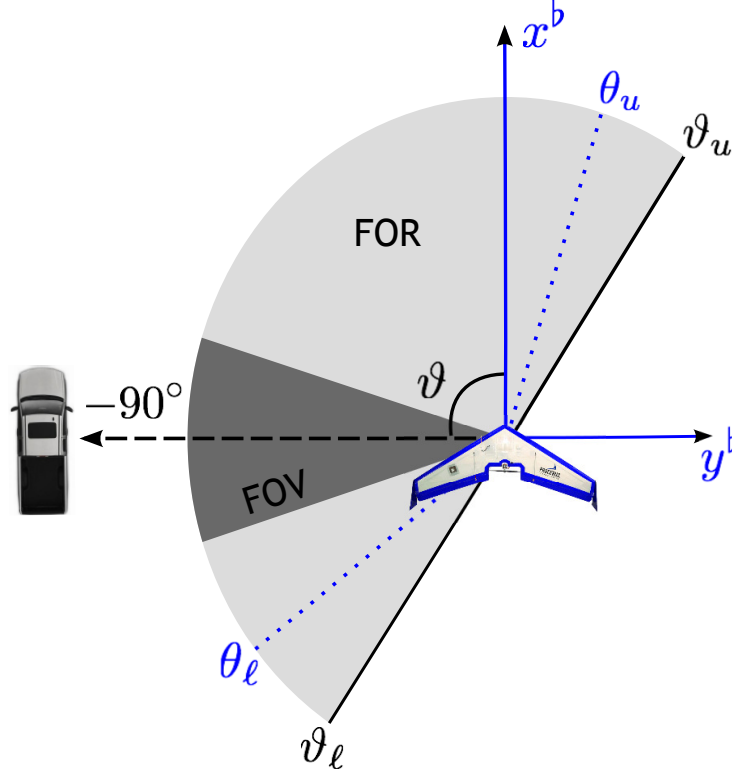
$$\mathcal{Z} := \mathbb{R}^2 \times \mathbb{S}^1 \times C \times \mathbb{R}_{\geq 0}, \quad (2.17)$$

The overall dynamics of the game,  $z_{k+1} = f(z_k, u_k, d_k)$ , are given implicitly by  $f_1(\xi_k, u_k)$  and  $f_2(\eta_k, d_k)$  and the preceding transformations of the states in (2.15) and (2.16). Note that since the fourth state is the roll angle  $r$  that takes on a finite number of discrete values, we sometimes refer to the state space as being “partially discrete.”

### 2.2.2 Cost Objective

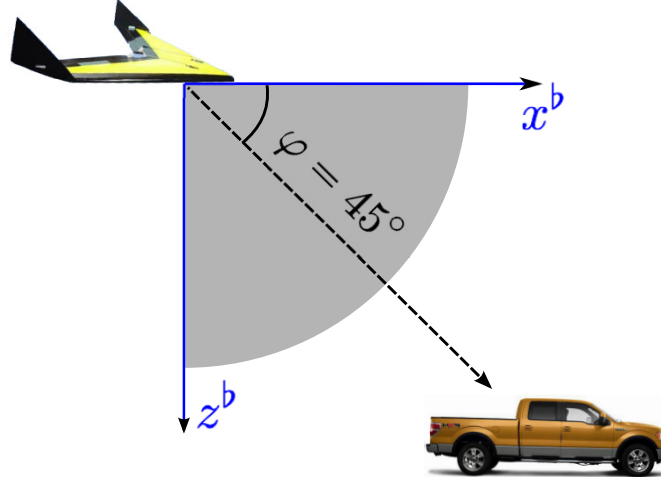
Small, inexpensive UAVs performing vision-based target tracking commonly carry miniature pan-tilt gimbal mechanisms that have limited sensing regions similar to the one depicted in Figure 2.2. The most prominent feature of this diagram is that there is a large blind spot extending from the right side of the UAV to its back, and hence a UAV with this particular field of regard would have to keep the target to its left for visibility.

Typical tilt angle limitations for a miniature pan-tilt gimbal mechanism are illustrated in Figure 2.3. While the mechanical tilt angle limitations shown in



**Figure 2.2:** The camera’s instantaneous *field of view* (FOV) and total *field of regard* (FOR) are indicated by the dark and light gray regions, respectively. The FOR is the total area visible to the camera as the gimbal is panned from its lower mechanical limit  $\theta_\ell$  to its upper mechanical limit  $\theta_u$ . The “b” superscript on each axis denotes the UAV’s local North-East-Down body frame. The azimuth angle of the line-of-sight vector to the target in this body frame is indicated by  $\vartheta$ , and if it lies within the upper and lower FOR bounds,  $\vartheta_u$  and  $\vartheta_\ell$ , respectively, then the target is in the UAV’s field of regard. Although the camera’s pan angle is not explicitly shown, it is assumed to equal the azimuth angle  $\vartheta$  when  $\vartheta \in [\theta_\ell, \theta_u]$  and otherwise be saturated at either  $\theta_\ell$  or  $\theta_u$ .

Figure 2.3 do not create blind spots in the down-looking direction, there are still reasons to avoid extreme tilt angles. In particular, a tilt angle close to  $0^\circ$  usually means that the UAV airframe is visible to the camera, which can block visibility of the target and/or generate false detections in image processing software. On



**Figure 2.3:** The range of tilt angles is indicated by the shaded region. The elevation angle of the line-of-sight vector to the target is denoted by  $\varphi$  and is measured in a positive sense from the  $(x^b, y^b)$  plane of the aircraft. For simplicity, the camera's tilt angle is taken to be the same as  $\varphi$ .

the other hand, a tilt angle close to  $90^\circ$  results in unpredictable movement of the gimbal, as this represents a singularity point in the gimbal geometry, i.e., the pan angle is not unique [7].

Based on these sensing limitations, which this work treats as soft constraints, the game objective for the UAV will be to maintain a good viewing geometry with respect to the ground target while the target's objective is the opposite. This means that the UAV wants to be as close to the target as possible while simultaneously avoiding extreme gimbal angles. This can be captured by a cost objective for the game to be minimized by the UAV and maximized by the target

given by

$$g(z) = g_1(\vartheta) + g_2(\varphi) + g_3(\rho), \quad (2.18)$$

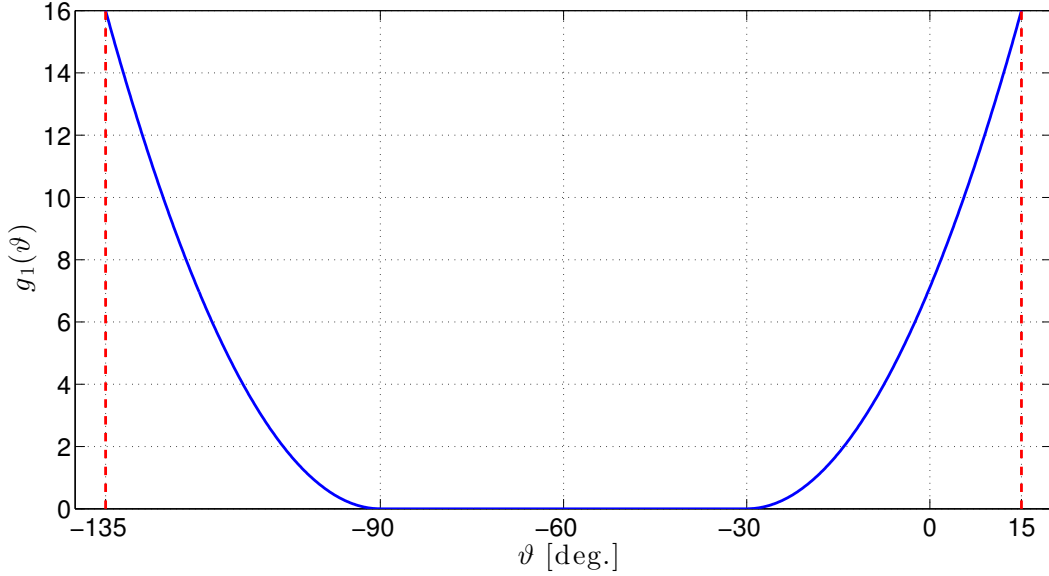
where  $\vartheta$  is the azimuth angle in Figure 2.2,  $\varphi$  is the elevation angle in Figure 2.3,  $\rho = \sqrt{(x_a - x_g)^2 + (y_a - y_g)^2}$ , and

$$\begin{aligned} g_1(\vartheta) &= \left( \lambda_1 \max\{\underline{\vartheta} - \vartheta, 0, \vartheta - \bar{\vartheta}\} \right)^2 \\ g_2(\varphi) &= \lambda_2^2 (\varphi - \pi/4)^2 \\ g_3(\rho) &= (\lambda_3 \rho)^2 \end{aligned} \quad (2.19)$$

with  $\lambda_i > 0$  and  $\bar{\vartheta} \geq \underline{\vartheta}$ . An example of  $g_1(\vartheta)$  is given in Figure 2.4 with  $\underline{\vartheta} = -90^\circ$ ,  $\bar{\vartheta} = -30^\circ$ , and  $\lambda_1 = 16\pi$ . The plot depicts the zero-cost interval of azimuth angles as well as the quadratic penalization that occurs as the azimuth angles leave this range and approach the extremities. Since the actions available to each player and cost objective have been described, the gameplay setup and associated dynamic programming solution are presented next.

### 2.2.3 Game Setup and Objective

Although the dynamics of each agent were introduced in continuous-time, each agent chooses its control action at discrete time steps and applies the control action over the  $T_s$ -second ZOH period. Thus, the two-player, zero-sum dynamic



**Figure 2.4:** The azimuth cost function  $g_1(\vartheta)$ . In this particular instance, the pan angle limitations,  $\theta_\ell$  and  $\theta_u$ , are indicated by the horizontal dashed lines at  $\vartheta = -135^\circ$  and  $\vartheta = 15^\circ$ , respectively.

game is played on a finite time interval of length  $K$  according to the discrete-time dynamics described in Section 2.2.1. At stage (time-step)  $k \in \{0, \dots, K - 1\}$ , the UAV's action  $u_k$  belongs to the action space  $U(r_k)$  defined in (2.7), and the target's control action pair  $d_k$  belongs to the action space  $D(v_k)$  defined in (2.14).

The game is played with a perfect state-feedback information structure, i.e., the players have access to the entire state, uncorrupted by noise, in order to decide on their actions. Furthermore, each player decides control actions according to a *behavioral policy*, which is a decision rule that associates to each state  $z \in \mathcal{Z}$  at stage  $k$  a probability distribution over the possible actions available to that player (see [17], Chapter 8). Therefore, when a player finds itself in a particular

state  $z \in \mathcal{Z}$ , it selects an action *randomly* according to the probability distribution given by the behavioral policy for  $z$ . The probability distributions over the UAV actions and target actions will belong to the following state-dependent probability simplexes:

$$\mathcal{A}(r) = \left\{ \alpha \in \mathbb{R}^{m(r)} : \sum_i \alpha_i = 1, \alpha_i \geq 0 \quad \forall i \right\} \quad (2.20)$$

$$\mathcal{B}(v) = \left\{ \beta \in \mathbb{R}^{n(v)} : \sum_j \beta_j = 1, \beta_j \geq 0 \quad \forall j \right\}, \quad (2.21)$$

respectively, where  $m(r) = |U(r)|$  is the number of actions available to the UAV and  $n(v) = |D(v)|$  is the number of action pairs available to the target. Accordingly, the behavioral policies for the UAV and target comprise time-dependent mappings  $\gamma_k : \mathcal{Z} \rightarrow \mathcal{A}(r)$  and  $\kappa_k : \mathcal{Z} \rightarrow \mathcal{B}(v)$ , respectively, and the control actions are realizations of state-dependent probability distributions defined by the behavioral policies:

$$u_k \sim \gamma_k(z), \quad d_k \sim \kappa_k(z), \quad \forall k \in \{0, \dots, K-1\}.$$

A particular behavioral policy of the UAV is denoted by  $\gamma$  and belongs to the action space  $\Gamma_1$ , which is the set of admissible behavioral policies for the UAV, i.e., the set of all length  $K$  sequences of mappings from  $\mathcal{Z}$  to  $\mathcal{A}(r)$ . Similarly, a particular behavioral policy for the target is denoted by  $\kappa$  and belongs to the action space  $\Gamma_2$ , which is the set of admissible behavioral policies for the target, i.e., the set of all length  $K$  sequences of mappings from  $\mathcal{Z}$  to  $\mathcal{B}(v)$ .

Because the state of the game evolves stochastically, the function for the UAV to minimize and the target to maximize is

$$J(\gamma, \kappa) = \mathbb{E} \left[ \sum_{k=0}^K g(z_k) \middle| z_0 \right],$$

where  $z_0 \in \mathcal{Z}$  and  $\mathbb{E}[\cdot]$  denotes expectation. Finally, the goal is to determine a saddle-point pair of behavioral policies  $(\gamma^*, \kappa^*)$  such that

$$J(\gamma^*, \kappa) \leq J(\gamma^*, \kappa^*) \leq J(\gamma, \kappa^*), \quad \forall \gamma \in \Gamma_1, \quad \forall \kappa \in \Gamma_2$$

for all initial conditions  $z_0 \in \mathcal{Z}$ . From the first condition, if the UAV is playing optimally with  $\gamma^*$ , the target (maximizer) can do no better (and may do worse) if it plays with any other admissible policy besides  $\kappa^*$ . And from the second condition, if the target is playing optimally with  $\kappa^*$ , the UAV (minimizer) can do no better (and may do worse) if it plays with any other admissible policy besides  $\gamma^*$ . Thus,  $(\gamma^*, \kappa^*)$  constitutes a *saddle-point* pair of *equilibrium* policies from which no player will deviate, lest it do strictly worse.

## 2.2.4 Dynamic Programming Solution

Dynamic programming can be used to determine the optimal policies  $\gamma^*, \kappa^*$  and the corresponding value of the game  $J(\gamma^*, \kappa^*)$  through *value iteration* as presented in Chapter 16 of [17]. This method centers on the *value function*  $V_k(z)$ , which is also referred to as the *cost-to-go* from  $z \in \mathcal{Z}$  at time  $k \in \{0, 1, \dots, K\}$ .



The value function is initialized at the final time as  $V_K(z) = g(z)$ . To determine the value function at the remaining times, an intermediate time and state-dependent matrix  $G(k, z) \in \mathbb{R}^{m(r) \times n(v)}$  is first introduced and is given by

$$[G_{ij}(k, z)] := g(z) + V_{k+1}(f(z, u^{(i)}, d^{(j)})), \quad (2.22)$$

where  $f(\cdot)$  refers to the dynamics of Section 2.2.1, and  $u^{(i)}$  and  $d^{(j)}$  are the  $i$ th and  $j$ th elements of the action spaces  $U(r)$  and  $D(v)$ , respectively. For  $k \in \{0, 1, \dots, K-1\}$ , the cost-to-go is computed (offline) in reverse chronological order according to the following recursion

$$\begin{aligned} V_k(z) &= \min_{\alpha \in \mathcal{A}(r)} \max_{\beta \in \mathcal{B}(v)} \alpha^\top G(k, z) \beta \\ &= \max_{\beta \in \mathcal{B}(v)} \min_{\alpha \in \mathcal{A}(r)} \alpha^\top G(k, z) \beta, \end{aligned} \quad (2.23)$$

where the second equality holds due to the Minimax Theorem (see [17], Chapter 5). As the optimizations in (2.23) are performed, the behavioral control policies are formed as

$$\gamma_k^*(z) = \arg \min_{\alpha \in \mathcal{A}(r)} \max_{\beta \in \mathcal{B}(v)} \alpha^\top G(k, z) \beta \quad (2.24)$$

$$\kappa_k^*(z) = \arg \max_{\beta \in \mathcal{B}(v)} \min_{\alpha \in \mathcal{A}(r)} \alpha^\top G(k, z) \beta, \quad (2.25)$$

and the procedure yields  $J(\gamma^*, \kappa^*) = V_0(z)$ ,  $\forall z \in \mathcal{Z}$ .

For computational feasibility, the computation of the value function is limited to a finite number of points through the introduction of  $Z \subset \mathcal{Z}$ , which is a finite

subset of the state space having  $N_s$  distinct elements. Computation of the value function is approximated by first redefining  $G(k, z)$  as

$$[G_{ij}(k, z)] := g(z) + V_{k+1} \left( q[f(z, u^{(i)}, d^{(j)}), Z] \right), \quad (2.26)$$

where  $q(\cdot)$  is given by (2.2) and then computing the value function according to (2.23) only for  $z \in Z$ . Similarly, the behavioral policies are computed for  $z \in Z$  according to (2.24) and (2.25), but with  $G(k, z)$  as defined in (2.26), instead of (2.22). In practice, the control policies for  $z \in \mathcal{Z} \setminus Z$  at time  $k$  are evaluated using  $\gamma_k^*(q(z, Z))$  and  $\kappa_k^*(q(z, Z))$ .

In general, determining (2.23) requires solving a linear program, which can be computationally burdensome for large  $N_s$ . However, when

$$\min_i \max_j G_{ij}(k, z) = \max_j \min_i G_{ij}(k, z), \quad (2.27)$$

where  $i \in \{1, 2, \dots, m(r)\}$  and  $j \in \{1, 2, \dots, n(v)\}$ , then one does not have to solve a linear program, as a saddle-point equilibrium exists in pure policies, and one can set

$$\gamma_k^*(z) = \mathbf{e}_i, \quad i := \arg \min_i \max_j G_{ij}(k, z) \quad (2.28)$$

$$\kappa_k^*(z) = \mathbf{e}_j, \quad j := \arg \max_j \min_i G_{ij}(k, z), \quad (2.29)$$

where  $\mathbf{e}_i$  and  $\mathbf{e}_j$  are the  $i$ th and  $j$ th vectors of the canonical basis of  $\mathbb{R}^{m(r)}$  and  $\mathbb{R}^{n(v)}$ , respectively.

The optimization for the hardware experiments was performed with the parameter data included in Tables 2.1, 2.2, and 2.3. Note that each state is uniformly sampled over a bounded interval (Table 2.3), and the discrete state space  $Z$  comprises the Cartesian product of the resulting discrete sets. This entails that no intuition regarding the nature of the optimal solution is exploited, e.g., discarding regions of the state space wherein the target is on the right-hand side of the UAV. Thus, the dynamic programming solution remains quite general, and, even for target tracking problems with no gimbal constraints, i.e., those in which  $\lambda_1 = \lambda_2 = 0$  in (2.19), the computational effort remains the same.

To gain insight into the nature of the corresponding control policy, consider a projection of the state space onto the  $(z_1, z_2)$ -plane with  $(z_3, z_4, z_5) = (0, 0, 8)$  in Figure 2.5. Thus, this picture depicts the control actions for any position of the UAV relative to that of the target when the relative heading between the UAV and target is zero, the UAV's discrete roll angle  $r$  is zero (hence  $\bar{r} = u$ ), and the target is traveling at 8 m/s, which is just under half of the UAV's speed. The line of states in the first quadrant that corresponds to mixed control policies and  $u = 0^\circ$  likely arises from the angular argument of  $g_1(\vartheta)$  in (2.19) being wrapped to the interval  $[-\pi, \pi)$ . Otherwise, the first quadrant primarily depicts the UAV beginning a left turn ( $u = -15^\circ$ ) in order to perform a counterclockwise loop around target. In the upper-leftmost parts of the first, second, and third

**Table 2.1:** Optimization Parameters

Parameter:	$\underline{\vartheta}$	$\vartheta$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$h_a$	$K$
Value:	$-\pi/2$	$-\pi/6$	$16/\pi$	$20/\pi$	$3/200$	170	15
Units:	rad.	rad.	rad. <sup>-1</sup>	rad. <sup>-1</sup>	m <sup>-1</sup>	m	-

quadrants, the UAV turns right ( $u = +15^\circ$ ), to get behind the target and to its right. There is also a significant portion of the state space in the third quadrant for which the UAV simply continues its course ( $u = 0^\circ$ ). Supposing the target holds its velocity and the UAV holds its course, the UAV will eventually cross over into the fourth quadrant due to the ground speed differential, wherein it begins turning left sooner if it is either too close or too far from the target. In any case, when  $z_1 \geq 56$  [m], the UAV needs to turn left in order to loop around the target in the counterclockwise direction. Lastly, Figure 2.5 illustrates the fact that a vast majority of the UAV control actions in the discrete state space  $Z$  are deterministic, as the white area corresponding to mixed policies is small. In fact, (2.27) holds for more than 95% of the states in  $Z$ , which greatly reduces the number of linear programs that need to be solved.

## 2.3 Stochastic Optimal Control Design

This section presents the stochastic optimal control approach to vision-based target tracking with a single small UAV, which is an attractive alternative to

**Table 2.2:** Parameters in UAV and Target Dynamics

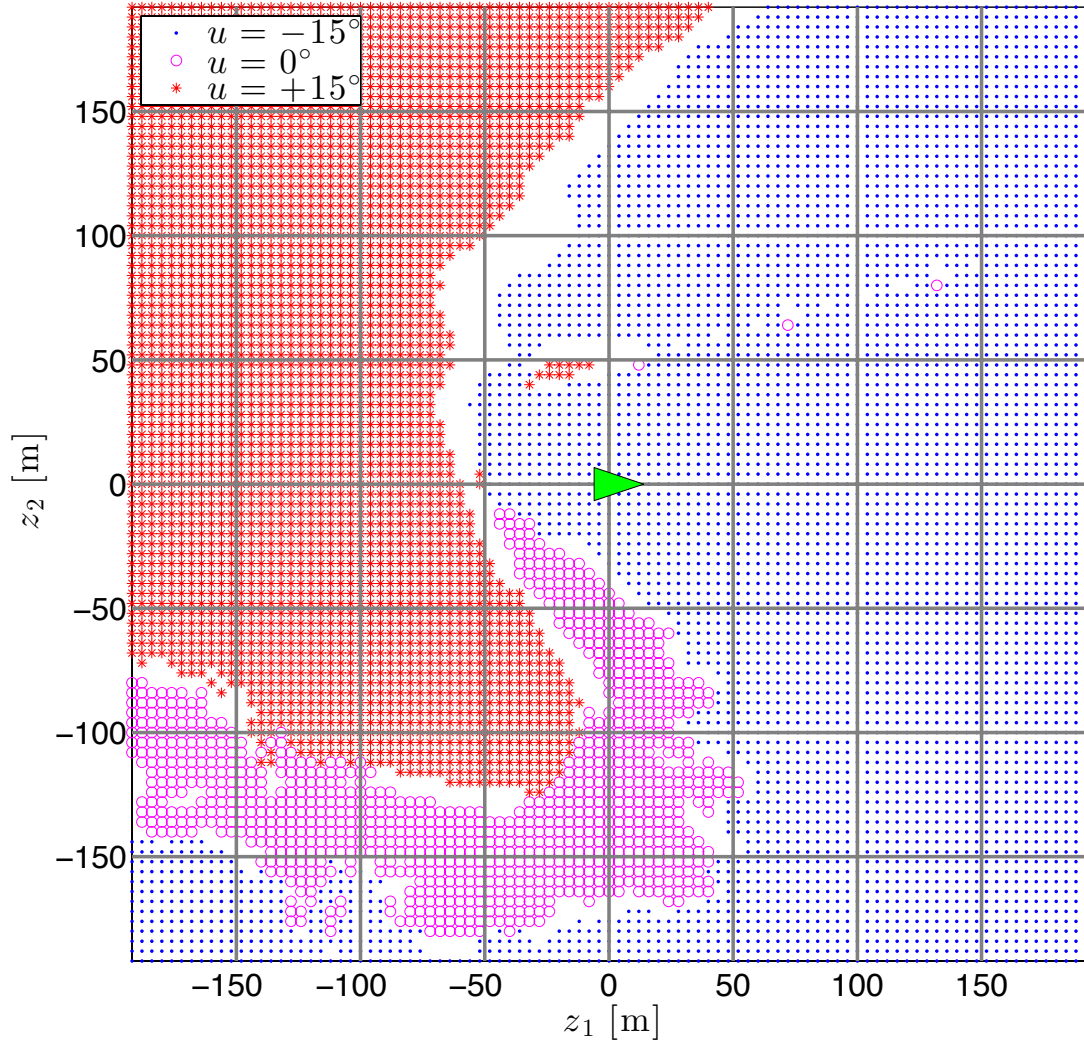
Parameter	Description	Value	Units
$s_a$	airspeed	18	m/s
$\alpha_g$	gravitational acceleration	9.81	m/s <sup>2</sup>
$\alpha_\phi$	inverse roll time constant	2	s <sup>-1</sup>
$\Delta$	max roll change	15	deg.
$\bar{\omega}$	max turn rate	$2\pi/24$	rad./s
$\bar{v}$	max target speed	14	m/s
$T_s$	zero-order hold period	2	s

**Table 2.3:** Sets for State Space Discretization

Set	Description	Value	Units
$X$	relative positions	$\{-192, -188, \dots, 192\}$	m
$\Psi$	relative headings	$\{0, 15, \dots, 345\}$	deg.
$C$	roll commands	$\{0, \pm 15, \pm 30\}$	deg.
$W$	target speeds	$\{0, 2, \dots, 14\}$	m/s
$Z$	discrete state space	$X^2 \times \Psi \times C \times W$	-

the computationally intensive game theoretic control design in Section 2.2. This approach utilizes more detailed models of the UAV and target to better characterize the actual dynamics encountered in a real-world implementation.

In Section 2.2, the UAV was modeled as a kinematic unicycle with first-order roll dynamics. In practice, the UAV's roll dynamics are far more complex, and the UAV's airspeed typically experiences significant stochasticity. To capture the effects of both of these observed characteristics, stochasticity is introduced into the UAV model without adding additional states beyond those encountered in the



**Figure 2.5:** Control surface for game theoretic control policy projected onto the  $(z_1, z_2)$  plane with  $(z_3, z_4, z_5) = (0, 0, 8)$ . Any points that appear to be missing are those for which the control action is not deterministic, i.e., the control policies are mixed and not pure. Note that the current roll-angle setpoint is  $\bar{r} = r + u = u$ , as  $r = 0^\circ$  in this particular case.

game theoretic control design. This leads to a stochastic optimal control problem that is solved over a finite planning horizon using dynamic programming.

### **2.3.1 Overview of Stochastic Dynamics**

The UAV and target states are assumed to evolve stochastically according to discrete-time Markov Decision Processes, whose states are described in Section 2.2.1. Accordingly, the probability of transitioning from the current UAV state  $\xi$  to the next UAV state  $\xi'$  under the change  $u$  in  $r$  is given by the controlled state transition probability function  $p_a(\xi' | \xi, u)$ . Likewise, the probability of transitioning from the current target state  $\eta$  to the next target state  $\eta'$  is given by the state transition probability function  $p_g(\eta' | \eta)$ .

Rather than deriving explicit formulas for these state transition probabilities, empirical characterizations of the individual agent kinematics are developed that allow one to draw Monte Carlo samples  $\tilde{\xi}^{(i,u)}$  and  $\tilde{\eta}^{(i)}$ ,  $i \in \{1, 2, \dots, N_p\}$ , from appropriate conditional probability density functions. (The quantity  $\tilde{\xi}^{(i,u)}$  denotes the sampled UAV state when control action  $u$  is applied to the UAV's initial state  $\xi$ .) To generate sample trajectories of the combined state  $z$ , one simply combines  $\tilde{\xi}^{(i,u)}$  and  $\tilde{\eta}^{(i)}$  according to equations (2.15) and (2.16). This provides an empirical characterization of the overall stochastic dynamics that evolve according to a controlled state transition probability function  $p(z' | z, u)$ . The ability to sample this state transition probability will suffice to effectively approximate the dynamic programming solution.

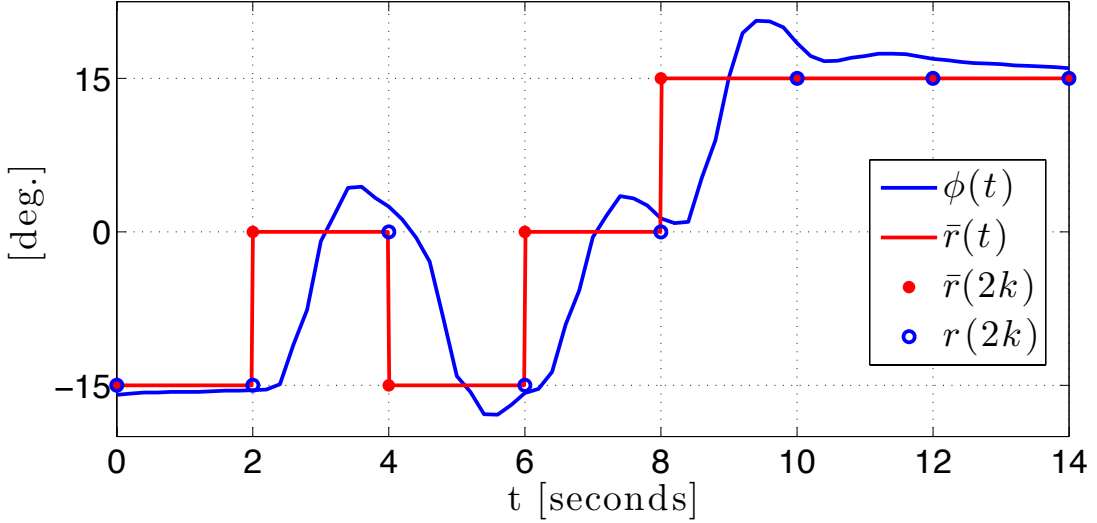
### **2.3.2 Stochastic UAV Kinematics**

In practice, UAVs are subject to environmental disturbances, such as wind gusts, that introduce stochasticity into the dynamics. Although a real UAV's kinematics are most accurately captured by an aircraft model with 6 degrees of freedom (a 6-DoF model), this work uses a 4-state stochastic model of the kinematics, in which stochasticity accounts for the effects of both unmodeled dynamics (arising from the reduced 4<sup>th</sup> order model) and environmental disturbances. The UAV state is the same as that in the game theoretic control design of Section 2.2.1.

To develop the stochastic discrete-time kinematic model, the deterministic continuous-time model of (2.10) is used, but with a much more detailed model for the controlled roll dynamics. As in the game theoretic case, a  $T_s$ -second zero-order hold (ZOH) is applied to the control input  $u$ , which again represents the change in the discrete roll angle  $r$ . Again, the maximum allowable change in roll angle is  $\Delta$ . Finally,  $r$  and  $u$  belong to the sets  $C$  and  $U(r)$  defined in (2.3) and (2.7), respectively.

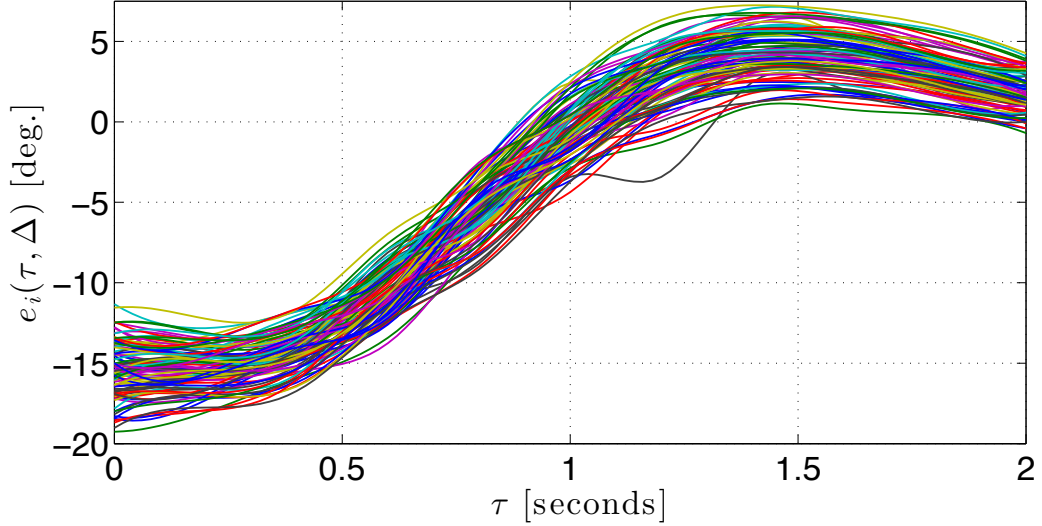
To draw Monte Carlo samples from an appropriate state transition probability function for the UAV for each possible roll action, roll trajectories are sampled using Monte Carlo simulations from the high-fidelity flight simulator Aviones [1], which utilizes a 6-DoF aircraft model. In particular, at each discrete





**Figure 2.6:** Monte Carlo simulations to sample roll trajectories. Once every  $T_s = 2$  seconds the discrete roll angle is randomly changed by  $u_k \in U(r_k)$ , where each element occurs with equal probability. Also, the maximum allowable change in the discrete roll angle  $r$  is  $\Delta = 15^\circ$ .

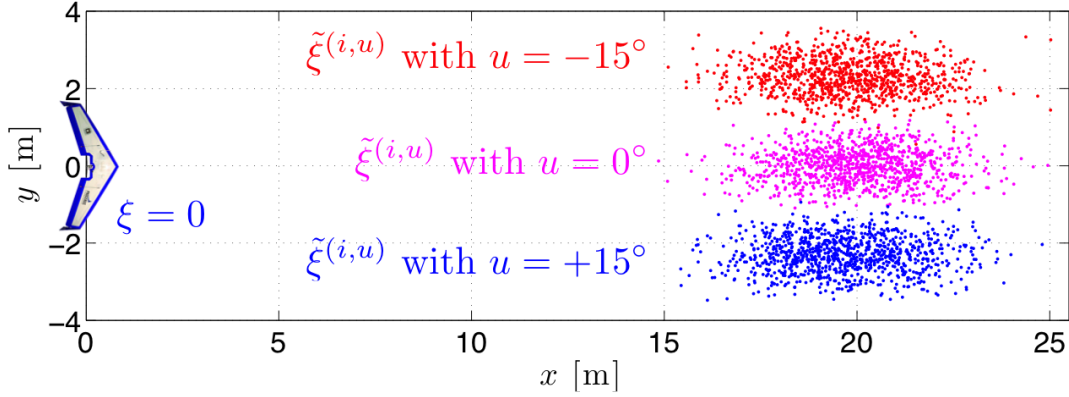
time instance of  $t = kT_s$  seconds, where  $k \in \mathbb{Z}_{\geq 0}$ , a control action  $u_k \in U(r_k)$  is selected randomly, where each element of  $U(r_k)$  occurs with equal probability and  $r_k = q(\phi_k, C)$ . The roll-angle setpoint  $\bar{r}(t) = r(t) + u(t)$  is held constant over each ZOH period, i.e.,  $\forall t \in [kT_s, kT_s + T_s)$ . This process (illustrated in Figure 2.6) is continued until families of  $N_p$  roll trajectories are collected for each possible change  $u \in \mathfrak{U}$ . Corresponding to this collection of roll trajectories is the collection of reference tracking error trajectories  $\{e_i(\tau, u)\}$ ,  $i \in \{1, 2, \dots, N_p\}$ , where each error trajectory takes the form  $e_i(\tau, u) = \phi(t) - \bar{r}(t)$  for some  $t \in \mathbb{R}_{\geq 0}$ , with  $\tau = t - \lfloor t/T_s \rfloor T_s$  and  $\lfloor \cdot \rfloor$  denoting the floor function. Since  $k = \lfloor t/T_s \rfloor$ , we have that  $\tau \in [0, T_s)$ . For  $\tau$  to be defined on the closed interval  $[0, T_s]$ , we take



**Figure 2.7:** One hundred error trajectories over a  $T_s = 2$  second ZOH period resulting from an increase of  $u = \Delta = 15^\circ$  in the discrete roll angle  $r$ . Note that each initial condition is  $e_i(0, u) = \phi(k_i T_s) - \bar{r}(k_i T_s)$  for some  $k_i \in \mathbb{Z}_{\geq 0}$ , and hence these initial conditions need not be zero.

$e_i(T_s, u) := \phi((k+1)T_s) - \bar{r}(kT_s)$ . Figure 2.7 shows a collection of reference tracking error trajectories,  $\{e_i(\tau, u)\}$ ,  $\tau \in [0, T_s]$ , over the  $T_s$ -second ZOH period corresponding to  $N_p = 100$  increases of  $u = \Delta$  in  $r$ .

The collection of these Monte Carlos simulations provides a large sample of roll-angle trajectories. In particular, to generate a sample roll trajectory  $\phi_i(\tau, r, u)$ ,  $\tau \in [0, T_s]$ , corresponding to an change of  $u$  in the discrete roll angle needed to reach the reference roll-angle setpoint  $\bar{r} \in C$ , one takes  $\phi_i(\tau, r, u) = e_i(\tau, u) + r + u$ . For example, if  $r = -15^\circ$  in Figure 2.7, then  $\phi_i(\tau, -15^\circ, 15^\circ) = e_i(\tau, 15^\circ)$ . Since  $\phi_i(T_s, -15^\circ, 15^\circ) < (\Delta/2) = 7.5^\circ$  for all  $i \in \{1, 2, \dots, N_p\}$ , we have  $q(\phi_i(T_s, -15^\circ, 15^\circ), C) = 0 = \bar{r}$ , which entails the roll angle approximately



**Figure 2.8:** Sample trajectories generated from the stochastic kinematic model for the UAV with  $s \sim \mathcal{N}(10, 16/25)$ ,  $T_s = 2$  seconds, and  $\Delta = 15^\circ$ . The initial UAV state is identically zero. For each  $u \in U(0^\circ) = \{0^\circ, \pm 15^\circ\}$ , 1,000 sample trajectories were generated. For each command, the vertical spread in final UAV positions is due to sampling different roll trajectories while the horizontal spread results from stochastic airspeed.

achieves the setpoint  $\bar{r} = r + u$  in each of the illustrated cases. In like manner, all of the roll trajectories used in the model approximately achieve the roll-angle setpoint at the end of the ZOH period; consequently,  $r_{k+1} = r_k + u_k$ , as in the case of first-order roll dynamics. In the sequel,  $\Phi(r, u)$  will denote the collection of all the sample roll-angle trajectories so generated for each roll angle  $r \in C$  and control action  $u \in U(r)$ . Moreover,  $\Phi(r, u) := \{\phi_i(\tau, r, u)\}$ , where  $i \in \{1, 2, \dots, N_p\}$  and  $\tau \in [0, T_s]$ .

To make the aircraft model more realistic, stochasticity is also introduced into the airspeed  $s_a$ , which is taken to be normally distributed about a nominal

value  $\bar{s}$ , with variance  $\sigma_s^2$ . Furthermore, the airspeed  $s_a$  in (2.10) is assumed to be constant over each ZOH period.

This modeling technique results in samples for the “next” state  $\tilde{\xi}^{(i,u)}$  at discrete time  $k+1$ , given the “current” state  $\xi$  at time  $k$  and change  $u$  in the discrete roll angle  $r$ . Specifically, the first three components of  $\tilde{\xi}^{(i,u)}$  are the implicit solution to (2.10) at the end of the  $T_s$ -second ZOH period with initial condition  $\xi$ ,  $s_a \sim \mathcal{N}(\bar{s}, \sigma_s^2)$ , and  $\phi(t, r, u) \in \Phi(r, u)$ ; the fourth component of  $\tilde{\xi}^{(i,u)}$  is deterministic and is simply  $r + u$ . The samples thus have two sources of randomness: stochasticity in the roll-angle dynamics captured by the collection of roll-angle trajectories  $\Phi(r, u)$  and stochasticity in the airspeed. Figure 2.8 illustrates the stochastic UAV model.

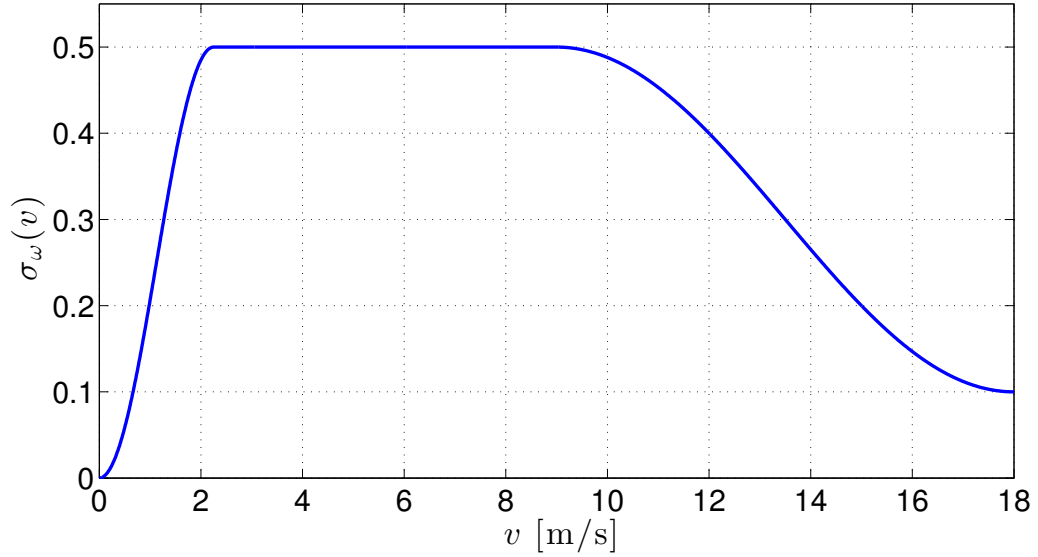
### 2.3.3 Stochastic Target Kinematics

To model the behavior of an operator driving the ground vehicle safely and casually, yet unpredictably, the same nonholonomic ground vehicle model given in Section 2.2.1 is used, where the continuous time dynamics are those given by (2.11) and each target control input is still held constant for  $T_s$  seconds. However, since evasive action is no longer assumed, control actions for the target are now drawn from continuous probability density functions that are independent of the UAV state.

While the target's acceleration is assumed to be a zero-mean Gaussian with fixed variance  $\sigma_a^2$ , its turn rate is generated from a more complex model. Specifically, the target's turn rate begins as a zero-mean Gaussian random variable whose variance depends on the current target speed  $v$  according to  $\sigma_\omega^2(v) = h^2(v/\bar{v})$ , where  $\bar{v}$  is the maximum target speed and

$$h(\nu) = \begin{cases} (\bar{\sigma}/2) [1 + \cos(\pi(\nu - \nu_1)/\nu_1)] & 0 \leq \nu < \nu_1 \\ \bar{\sigma} & \nu_1 \leq \nu < \nu_2 \\ \frac{\bar{\sigma} - \sigma_f}{2} \left[ 1 + \cos\left(\pi \frac{\nu - \nu_2}{1 - \nu_2}\right) \right] + \sigma_f & \nu_2 \leq \nu < 1, \end{cases}$$

which is an asymmetric cosine-tapered window with maximum value  $\bar{\sigma}$ , final value  $\sigma_f$ , and parameters  $\nu_1$  and  $\nu_2$  describing the beginning and end points of the window with  $0 < \nu_1 < \nu_2 < 1$ . This is illustrated in Figure 2.9 with the parameters included in Table 2.4. The fact that  $\sigma_\omega^2(0) = 0$  captures the assumption that the target is a nonholonomic vehicle that does not turn in place. Furthermore, as the target's speed increases from zero, it makes sharper turns more frequently, and there is even a range over which this frequency of sharp turns remains constant. Beyond this range, as its speed increases, the target makes softer turns in order to avoid losing traction/control. Thus, the normally distributed stochastic turn rate with speed-dependent variance models safe casual driving of the ground vehicle.



**Figure 2.9:** The speed dependent standard deviation of the normally distributed turn rate.

**Table 2.4:** Stochastic target motion parameters

Parameter:	$\sigma_a$	$\bar{v}$	$\bar{\sigma}$	$\sigma_f$	$\nu_1$	$\nu_2$	$\varrho$
Value:	0.5	18	0.5	0.1	1/8	1/2	7
Units:	m/s <sup>2</sup>	m/s	rad./s	rad./s	-	-	m

To add another degree of realism, the zero-mean Gaussian random variable with variance  $\sigma_\omega^2(v) = h^2(v/\bar{v})$  used to generate the turn rate  $\omega$  is “saturated” to enforce a minimum turn radius  $\varrho$ . Specifically, suppose that an acceleration sample  $a_i$  and turn rate sample  $\tilde{\omega}_i$  are drawn from  $\mathcal{N}(0, \sigma_a^2)$  and  $\mathcal{N}(0, \sigma_\omega^2(v))$ ,

respectively. The turn rate  $\omega_i$  that is actually used is given by

$$\omega_i = \begin{cases} \bar{\omega}_i, & \tilde{\omega}_i > \bar{\omega}_i \\ \tilde{\omega}_i, & -\bar{\omega}_i \leq \tilde{\omega}_i \leq \bar{\omega}_i \\ -\bar{\omega}_i, & \tilde{\omega}_i < -\bar{\omega}_i \end{cases}$$

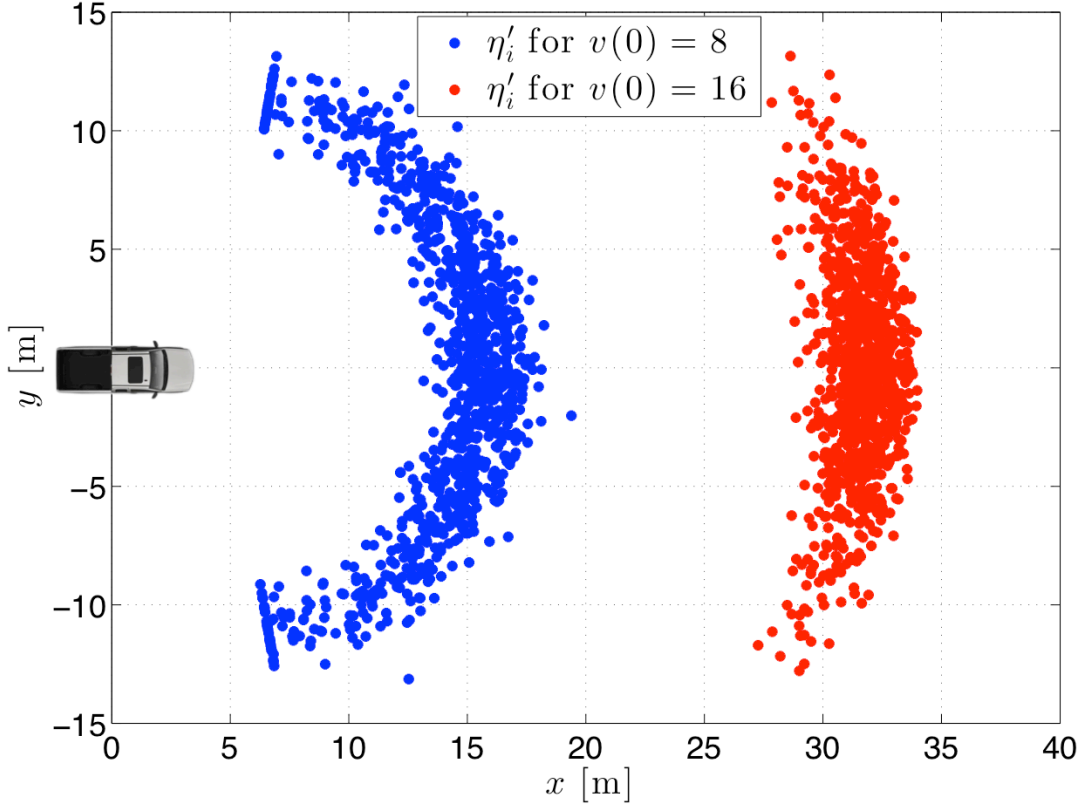
where  $\bar{\omega}_i := \min\{v'_i/\varrho, v/\varrho\}$  is the maximum turn rate consistent with a minimum turn radius  $\varrho$ , and  $v'_i = a_i T + v$  is the target speed at the end of the ZOH period. Figure 2.10 illustrates the stochastic target model with the parameters of Table 2.4 for two different initial conditions. In this figure, the scenario for the slower initial target speed has a final position distribution with a higher radius of curvature, which entails that the target makes sharper turns. The saturation on the tails of the turn-rate distribution becomes evident as well.

### 2.3.4 Control Objective and Dynamic Programming Solution

The stochastic optimal control problem is to determine the optimal control feedback control policy  $\mu_k^* : \mathcal{Z} \rightarrow \mathfrak{U}$ ,  $k \in \{0, 1, \dots, K-1\}$ , that minimizes

$$J(z_0) = \mathbb{E} \left[ \sum_{k=0}^K g(z_k) \middle| z_0 \right], \quad \forall z_0 \in \mathcal{Z}, \quad (2.30)$$

where  $K \in \mathbb{N}$ ,  $\mathbb{E}[\cdot]$  denotes expectation,  $g(z)$  is given by (2.18), and the state  $z$  is a Markov Decision Process that evolves according to the transition probability



**Figure 2.10:** Sample trajectories generated from the stochastic target motion model. The two initial target states depicted with different colors correspond to identical initial positions at the origin, but two distinct initial speeds of 8 and 16 [m/s]. For each initial condition,  $N_p = 1,000$  samples are generated.

$p(z' | z, u)$  determined by the models in Section 2.3.2 and Section 2.3.3, under the feedback law  $u_k = \mu_k^*(z_k)$ . Note that the state transition probability  $p(z' | z, u)$  can also be written as  $p(z_{k+1} | z_k, u_k)$ .

Similar to the first control design, dynamic programming is used to minimize the criterion (2.30) using *value iteration* as presented in [44]. The method hinges on the *value function*, or *cost-to-go* from state  $z \in \mathcal{Z}$  at time  $k \in \{0, 1, \dots, K-1\}$ ,



which is defined as

$$V_k(z) := \min_{u_k, u_{k+1}, \dots, u_{K-1}} \mathbb{E} \left[ \sum_{\ell=k}^K g(z_\ell) \middle| z_k = z \right].$$

For  $k = K$ , one takes  $V_K(z) = g(z)$ , and the cost-to-go for  $k \in \{0, 1, \dots, K-1\}$  is computed (offline) in reverse chronological order according to the following recursion

$$\begin{aligned} V_k(z) &= g(z) + \min_{u \in U(r)} \mathbb{E} [V_{k+1}(z') | z, u] \\ &= g(z) + \min_{u \in U(r)} \int V_{k+1}(z') p(z' | z, u) dz', \end{aligned} \quad (2.31)$$

which holds due to Bellman's principle of optimality (see [25], Chapter 6). As the minimization is performed, the optimal control policy is also formed as

$$\mu_k^*(z) = \arg \min_{u \in U(r)} \left( g(z) + \mathbb{E} [V_{k+1}(z') | z, u] \right). \quad (2.32)$$

Performing the sequence of computations in (2.31) for  $k \in \{K-1, K-2, \dots, 0\}$  ultimately yields  $J^*(z) = V_0(z)$ ,  $\forall z \in \mathcal{Z}$ , where  $J^*(z)$  is the minimum value of (2.30) under the feedback law (2.32). The main hurdle in computing (2.31) is the expectation, i.e., the integral over the implicitly specified state transition probability  $p(z' | z, u)$ , which is overcome in this work through a replacement with an empirical average computed using samples drawn according to the stochastic UAV and target models developed in Sections 2.3.2 and 2.3.3:

$$V_k(z) \approx g(z) + \min_{u \in U(r)} \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(\tilde{z}^{(i,u)}), \quad (2.33)$$

where  $\tilde{z}^{(i,u)} \sim p(z' | z, u)$ . As in Section 2.2.4, the computation of the value function is limited to a finite number of points in a set  $Z \subset \mathcal{Z}$  that has  $N_s$  elements. Accordingly, with the quantization function  $q(\cdot)$  from (2.2), the computation of the value function and optimal policy is approximated by

$$V_k(z) \approx g(z) + \min_{r \in U(r)} \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(q(\tilde{z}^{(i,u)}, Z))$$

$$\mu_k^*(z) = \arg \min_{r \in U(r)} \left( g(z) + \frac{1}{N_p} \sum_{i=1}^{N_p} V_{k+1}(q(\tilde{z}^{(i,u)}, Z)) \right),$$

which only requires the evaluation of the value function over the finite set  $Z$ . In practice, the optimal control action  $u_k$  for a state  $z \in \mathcal{Z} \setminus Z$  is determined using  $u_k = \mu_k^*(q(z, Z))$ .

Dynamic programming was performed for the hardware experiments with the parameters given in Tables 2.1, 2.4, 2.5, and 2.6, where  $N_p = 1,000$  sample points were used to approximate the expectation in (2.31) and (2.32). Note that, as in the solution for the game theoretic approach, each state is uniformly sampled over a bounded interval, and the overall discrete state space  $Z$  comprises the Cartesian product of the resulting discrete sets, as indicated in Table 2.6. Thus, no intuition into the nature of the optimal solution is exploited, and the dynamic programming solution to the stochastic optimal control problem remains quite general, requiring no additional computational effort even when there are no gimbal constraints, i.e., when  $\lambda_1 = \lambda_2 = 0$  in (2.19).

To understand the nature of the resultant UAV control policy, consider a projection of the state space onto the  $(z_1, z_2)$ -plane with  $(z_3, z_4, z_5) = (0, 0, 8)$ , which is shown in Figure 2.11. In the first quadrant, roughly below the line  $z_2 = (13/27)z_1 + 117$  (dashed), the UAV begins turning left ( $u = -15^\circ$ ) to perform a counterclockwise loop about the target, which is a costly maneuver as the distance will increase significantly during this act. However, if the UAV is above the said line, it rather begins turning right ( $u = +15^\circ$ ) to move behind the target and to its right. The UAV does this same maneuver if it is in the left half of the second quadrant or the upper left corner of the third quadrant. Furthermore, the large region in the third quadrant wherein the UAV maintains a straight course ( $u = 0^\circ$ ) extends neither below  $z_2 = -154$  nor to the right of  $z_1 = 0$ , which is not true for the game theoretic control policy  $\gamma^*$ . Since the UAV turns left below  $z_2 = -154$ , it is prevented from straying too far from the target. Supposing the UAV holds its course in the third quadrant and the target maintains its velocity, the UAV will cross into the fourth quadrant, where it begins turning left immediately to perform a counterclockwise turn about the target. This characteristic results in the UAV turning sooner than in the first control design and results in noticeably different behavior, which will become evident in the flight test results.

**Table 2.5:** Parameters in Stochastic UAV dynamics

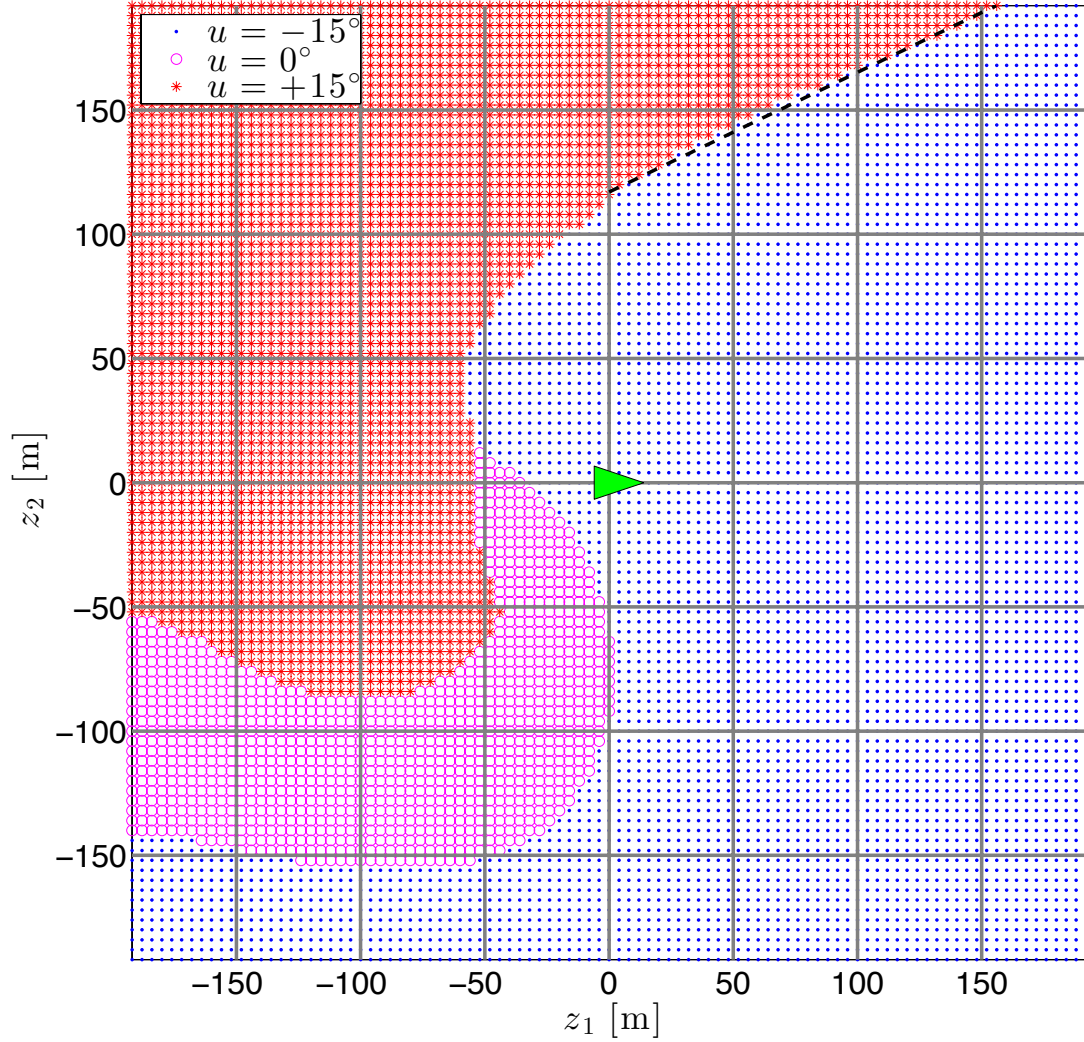
Parameter	Description	Value	Units
$\bar{s}$	nominal airspeed	18	m/s
$\sigma_s^2$	airspeed variance	16/25	m <sup>2</sup> /s <sup>2</sup>
$\alpha_g$	gravitational acceleration	9.81	m/s <sup>2</sup>
$\Delta$	max roll change	15	deg.
$T_s$	zero-order hold period	2	s

**Table 2.6:** Sets for State Space Discretization

Set	Description	Value	Units
$X$	relative positions	$\{-252, -248, \dots, 252\}$	m
$\Psi$	relative headings	$\{0, 15, \dots, 345\}$	deg.
$C$	roll commands	$\{0, \pm 15, \pm 30\}$	deg.
$W$	target speeds	$\{0, 1, \dots, 17\}$	m/s
$Z$	discrete state space	$X^2 \times \Psi \times C \times W$	-

## 2.4 Hardware Setup and Flight Test Results

This section presents field test results of a single camera-equipped UAV performing vision-based target tracking with the control policies described in Sections 2.2 and 2.3. It begins with a description of the hardware setup for the field experiments whereby both the game theoretic and stochastic optimal control policies were tested. Next, it highlights representative flight test results to illustrate the behavior and performance of each control policy. Finally, the section concludes with a quantitative comparison of both control policies to objectively determine the preferred method for a real-world implementation.



**Figure 2.11:** Control surface for stochastic optimal control policy projected onto the  $(z_1, z_2)$  plane with  $(z_3, z_4, z_5) = (0, 0, 8)$ .

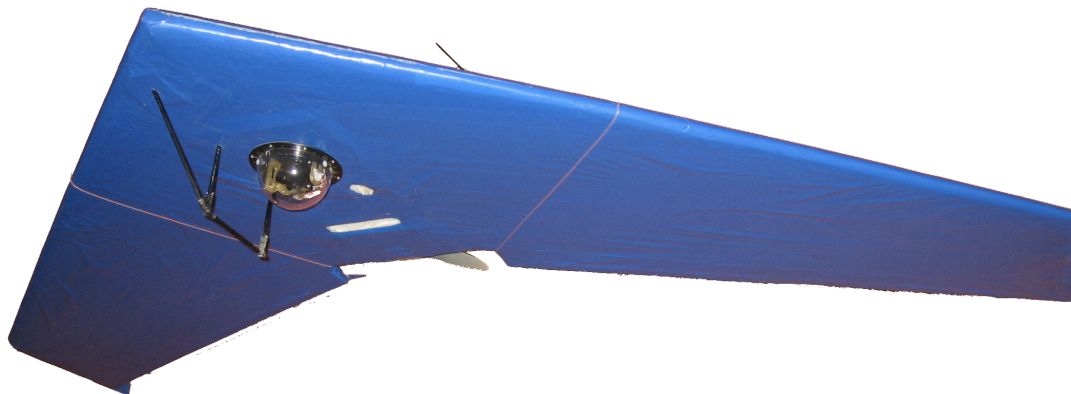
### 2.4.1 Experimental Setup

A flight experiment was conducted in February, 2012 with a single UAV at the McMillan Airfield, Camp Roberts, CA. Toyon Research Corporation was

responsible for launching, monitoring, and landing the UAV as well as driving the target ground vehicle.

A Unicorn/Zagi flying wing was used to test the vision-based target tracking control policies. This particular UAV platform is depicted in Figure 2.12 with its main payload, which is a gimbaled video camera. The particular pan and tilt limitations are those described in Figures 2.2 and 2.3, respectively, wherein the particular pan limits of the gimbal mechanism are  $\theta_\ell = -135^\circ$  and  $\theta_u = 15^\circ$ . The corresponding FOR limits are  $\vartheta_\ell = -152^\circ$  and  $\vartheta_u = 32^\circ$ . Note that throughout this section the azimuth angle  $\vartheta$ , and consequently the viewing geometry cost  $g(z)$ , is computed using the continuous roll angle  $\phi$  reported by the autopilot rather than the discretized roll angle  $r$ .

The UAV's autopilot adjusts the throttle and two elevons to maintain air-speed, altitude, and the commanded roll angle. The particular ZOH time of  $T_s = 2$  seconds was chosen to be roughly the settling time for the Unicorn's roll dynamics. The number of discrete time steps in each dynamic optimization was chosen to make the planning horizon ( $KT_s = 30$  seconds) longer than the time the UAV would spend in an orbit around the target ( $\approx 20$  seconds) at max bank. Furthermore, since both types of control policies were tested on the same sortie for a minimum of 15 minutes each, the limited planning horizon of 30 seconds was addressed by applying the control policies in a *receding horizon* fashion.



**Figure 2.12:** The Unicorn comprises a 60" expanded polypropylene foam wing that houses four lithium polymer batteries driving an electric motor attached to a push propeller.

This entails that, in the game theoretic case,  $u_k \sim \gamma_0^*(z)$ ,  $\forall k \in \mathbb{Z}_{\geq 0}$ , and in the stochastic optimal control case,  $u_k = \mu_0^*(z)$ ,  $\forall k \in \mathbb{Z}_{\geq 0}$ . In this manner, the UAV considered the impact its control decisions had on the viewing geometry(2.18) up to 30 seconds into the future, which proved beneficial for good long term performance.

During the tracking experiment, a human operator drove the ground vehicle casually, yet unpredictably, along the dirt roads of Camp Roberts, while the UAV was controlled by a ground control station (GCS). This is illustrated in Figure 2.13. Although the target remained on the roads for the duration of the tracking experiments, this did not benefit the tracking algorithm since the UAV's control policies were computed with no information regarding road networks. The UAV's control actions were computed using MATLAB<sup>®</sup>, which

communicated with the GCS and the GPS receiver onboard the truck to acquire the UAV telemetry and target data and to determine the roll command according to the particular control policy being tested on the UAV. The roll command was then sent back to the GCS and relayed to the UAV.

In real-world conditions, steady winds are often encountered having speeds that constitute a significant portion of a small UAV's airspeed. While the policies presented do not address heavy winds that alter the UAV's kinematics (projected onto the ground) significantly, light to moderate winds can be addressed in an approximate manner by altering the target's apparent ground velocity. In particular, Saunders notes in [41] that a constant target velocity and steady wind can be generalized to just a steady wind. We take a similar approach and combine the wind and target velocity to form the target's apparent ground velocity, since the coordinate frame defined by (2.15) and (2.16) is centered on the target. More specifically, denoting the wind by  $w \in \mathbb{R}^2$ , one can take the target's apparent heading  $\hat{\psi}_g$  and speed  $\hat{v}$  to be the following:

$$\hat{\psi}_g = \text{atan2}(v \sin \psi_g - w_2, v \cos \psi_g - w_1) \quad (2.34)$$

$$\hat{v} = \sqrt{(v \cos \psi_g - w_1)^2 + (v \sin \psi_g - w_2)^2}, \quad (2.35)$$

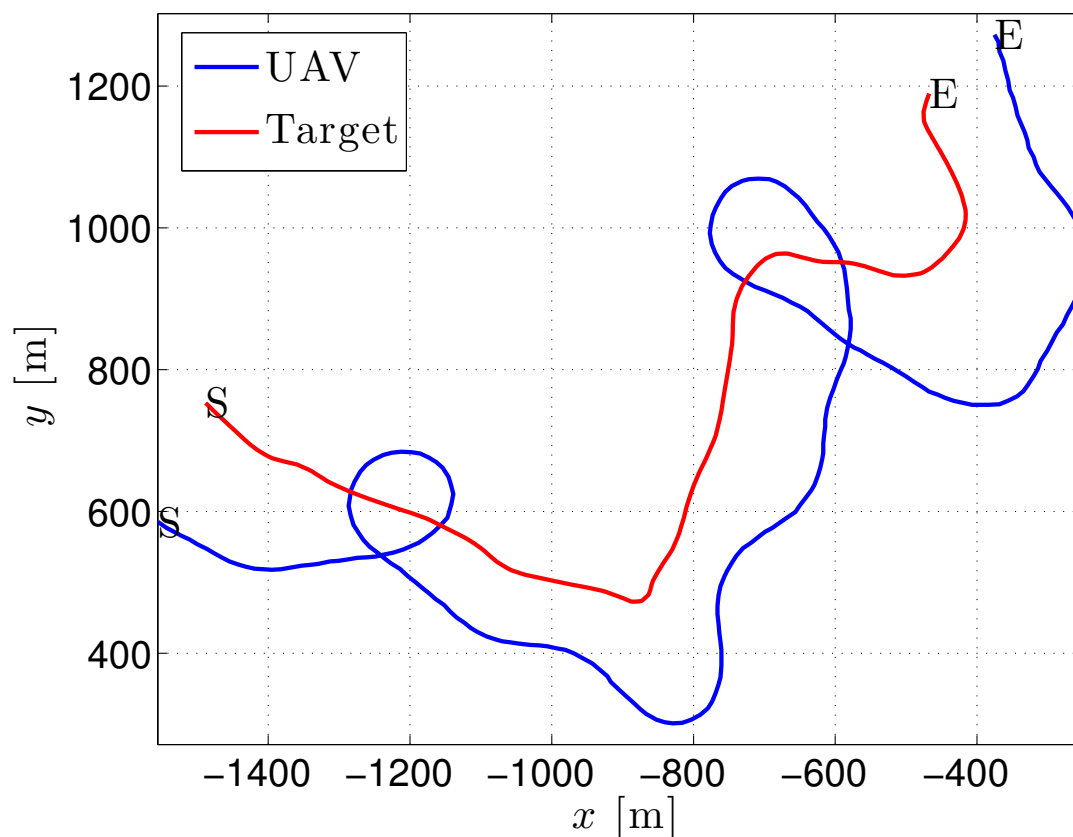
where  $\text{atan2}$  is the four-quadrant inverse tangent function. One then uses  $\hat{\psi}_g$  and  $\hat{v}$  in place of  $\psi_g$  and  $v$  in (2.15) and (2.16) to determine  $z$ .





**Figure 2.13:** A single UAV tracks the unpredictable ground target at Camp Roberts, CA with one of the optimized-based control policies. This image was captured by an independent UAV flying at a higher altitude and illustrates the typical imagery sent to the ground control station for video processing.

While this approximation was employed during the flight tests, its effects were quite negligible, as the average wind speed (as measured by the UAV) was less than  $0.5 \text{ [m/s]}$  over each 15-minute experiment. Since wind speeds are often greater than those that were experienced during this particular experiment, we address the issue of the control policies' robustness to steady winds at the end of the section and keep our focus on robustness to unpredictable target motion.



**Figure 2.14:** Trajectory of the UAV over 3 minutes as it tracks the target with the game theoretic control policy. An “S” indicates a starting position and “E” an ending position.

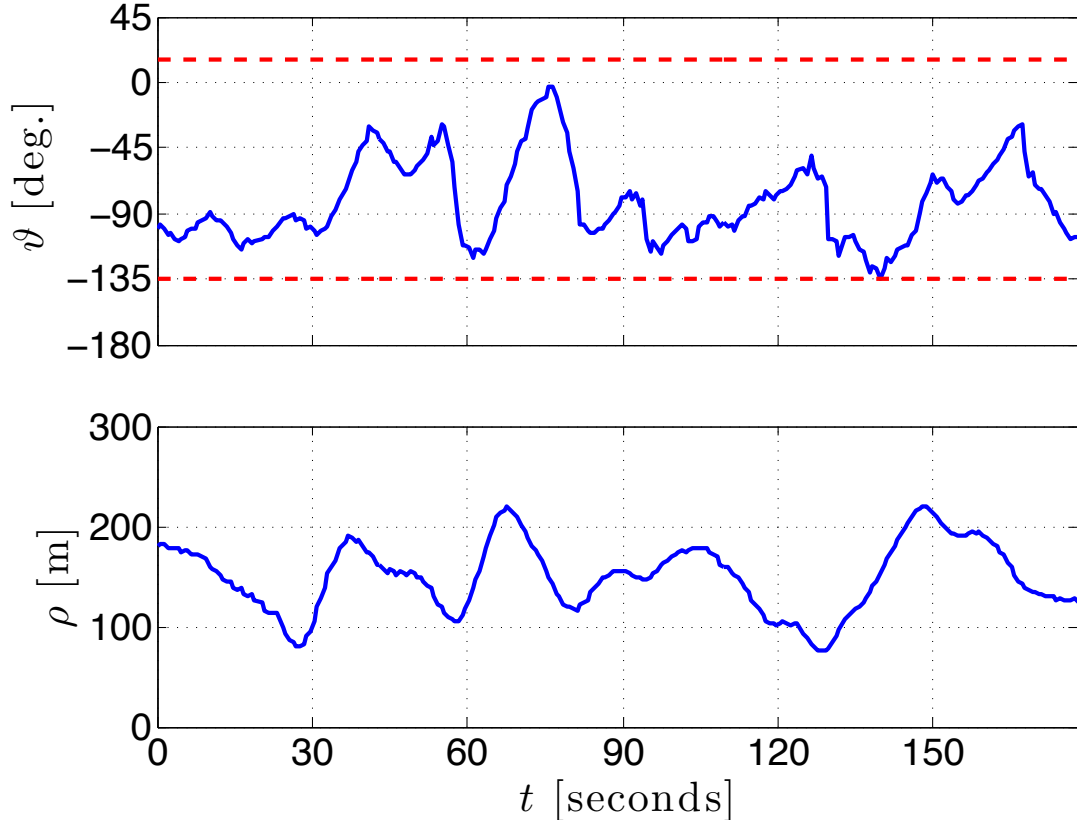
## 2.4.2 Game Theory Results

Three representative minutes of flight are now presented to highlight the key features of vision-based target tracking with the game theoretic control policy  $\gamma^*$ . The target and UAV trajectories during this window are provided in Figure 2.14. In the figure, the UAV keeps to the right of the target and occasionally makes counterclockwise turns about the target. Such behavior enables the UAV to

stay close to the target in spite of the speed differential and simultaneously keep the target in its FOR. Another characteristic of the UAV's trajectory is that it never passes over the target, which would minimize the distance to the target but lead to an elevation angle close to  $90^\circ$ . Such behavior illustrates the tradeoff between minimizing the distance to the target and avoiding a large elevation angle. Overall, the UAV exhibits standoffish behavior with the game theoretic control policy, meaning that it appears reluctant to perform loops that would likely make it vulnerable to an evasive target.

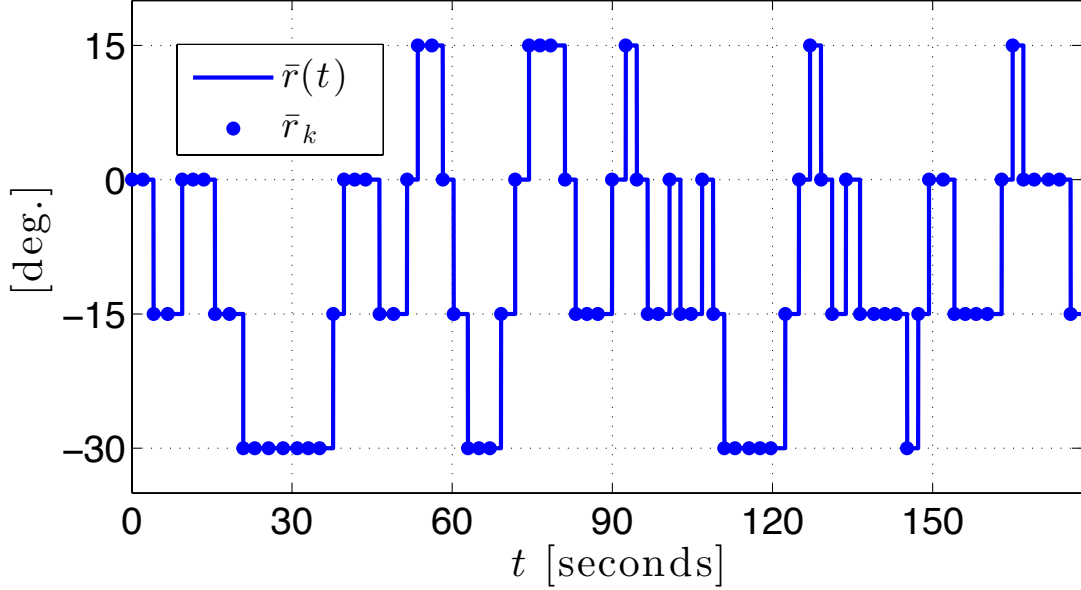
Figure 2.15 highlights the critical components of the viewing geometry. The azimuth angle was kept within the mechanical limits of the gimbal mechanism, and hence the target was kept within the FOV during this entire 3 minute window. However, the azimuth varies significantly in this window, as it takes on values near both extremes. This is quite visible in the time interval from  $t = 60$  to  $t = 75$  seconds when the gimbal slews from the back left of the UAV to the front. The gimbal also nearly hit its lower mechanical limit near  $t = 140$  seconds. Another feature of the viewing geometry is that the planar distance to target oscillates a fair amount, as it varies between roughly 100 and 200 meters during this 3-minute window.

Lastly, Figure 2.16 illustrates the roll command sequence under the game theoretic policy  $\gamma^*$ . One immediately apparent characteristic of this plot is



**Figure 2.15:** Critical components of the viewing geometry performance with the game theoretic control policy: azimuth  $\vartheta$  and 2-D distance to target  $\rho$ . The mechanical limits of the gimbal,  $\theta_\ell$  and  $\theta_u$ , are indicated by dashed lines in the plot of azimuth angle.

that the maximum roll command of  $+30^\circ$  is never used; however, the positive roll command of  $15^\circ$  is employed to achieve the standoffish behavior shown in Figure 2.14. Note how the roll-angle action space of (2.7) limits changes in roll to  $\Delta = 15^\circ$  degrees, and hence chattering between roll commands of opposite polarity is avoided.

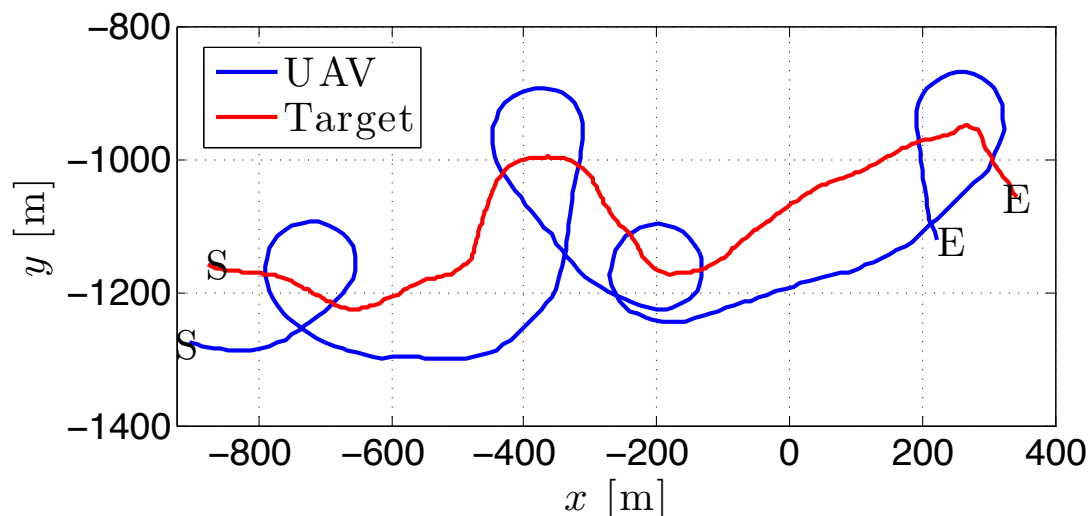


**Figure 2.16:** Roll command sequence under the game theoretic control policy  $\gamma^*$ .

### 2.4.3 Stochastic Optimal Control Results

Three representative minutes of flight during which roll commands were governed by  $\mu^*$  are now presented to highlight the salient features of vision-based target tracking with the stochastic optimal control policy  $\mu^*$ . The UAV and target trajectories are depicted in Figure 2.17. As with the game theoretic controller, the UAV keeps to the right of the target, and performs loops as necessary to compensate for the speed differential between itself and the target vehicle.

One immediate difference between the two control policies is that during the same time window the UAV performed twice as many loops under the stochastic optimal control policy. One might attribute this noticeable difference to the



**Figure 2.17:** Trajectory of the UAV over 3 minutes as it tracks the target with the stochastic optimal control policy. An “S” indicates a starting position and “E” an ending position.

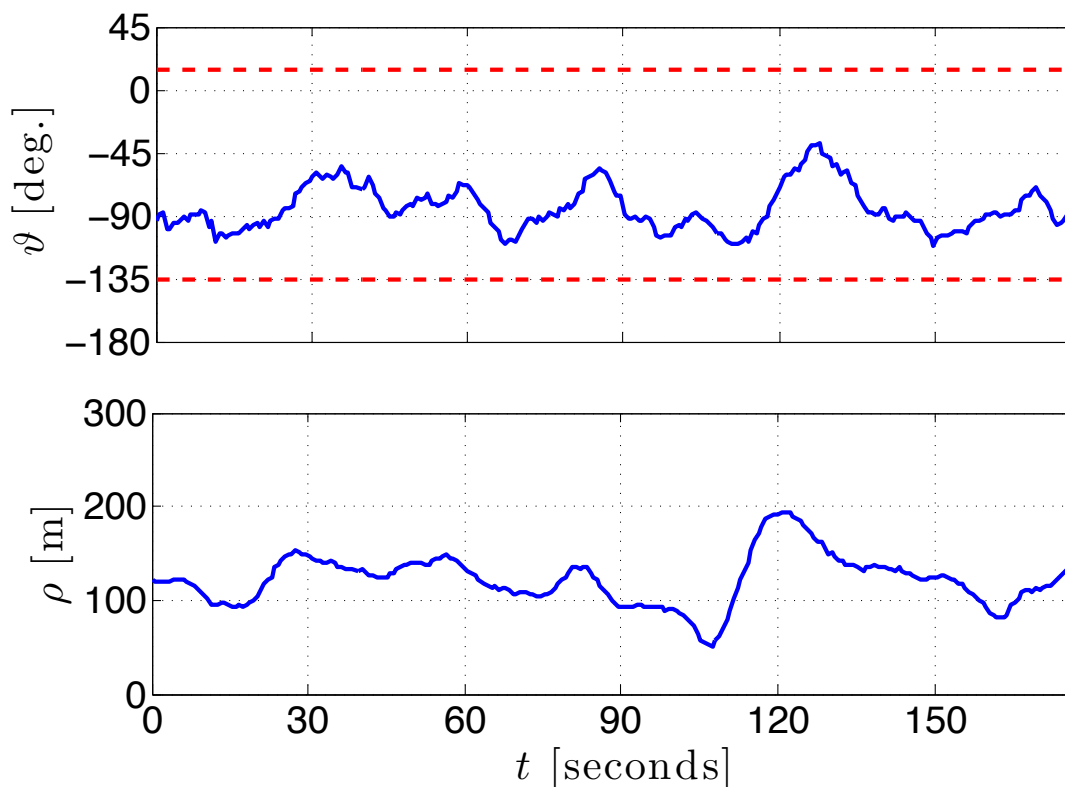
target speed profile over the 3 minute window for both cases; however, the target traveled nearly 1,739 [m] in the game theoretic case and 1,562 [m] in the stochastic optimal control case at average speeds of about 9.77 [m/s] and 8.83 [m/s], respectively. The average speed difference of just under 1 [m/s] is unlikely the primary source of this feature. Rather, this behavioral differences is best explained by the differences in the control policies themselves. In particular, as noted earlier in the discussion of the stochastic optimal control policy, the UAV makes loops earlier when it is traveling to the right of the target with zero roll and zero relative heading, as shown in the fourth quadrant of the control surface in Figure 2.11. With the game theoretic control policy, the UAV takes

a more precautionary stance against the target (assumed to be evasive), and accordingly, exhibits a reluctance to perform loops around the target.

Figure 2.18 illustrates the critical components of the viewing geometry, namely the azimuth angle and planar distance to target. One immediately noticeable feature of the azimuth angle  $\vartheta$  is that it oscillates very little in comparison to that of the game theoretic case. Furthermore, the gimbal never slews from one extreme to the next, nor does it come close to either extreme.

Concerning the planar distance to target  $\rho$ , there is one instance where it reaches nearly 200 [m], yet for the majority of the time window, the distance oscillates between 100 and 150 [m]. Moreover, the amplitude of the oscillations is nearly half that of the game theoretic case, and both the average value and variance of  $\rho$  are also significantly smaller.

Finally, Figure 2.19 depicts the roll command sequence under the stochastic optimal control policy  $\mu^*$ . The most prominent feature of this figure is that no positive roll commands are ever employed. Rather, one will notice a slight chattering behavior between  $\bar{r} = 0^\circ$  and  $\bar{r} = -15^\circ$ . Also, during this particular 3-minute time window, anytime the roll command  $\bar{r} = -30^\circ$  degrees is used the UAV is performing a left turn, which cannot be said of the game theoretic control policy  $\gamma^*$ .

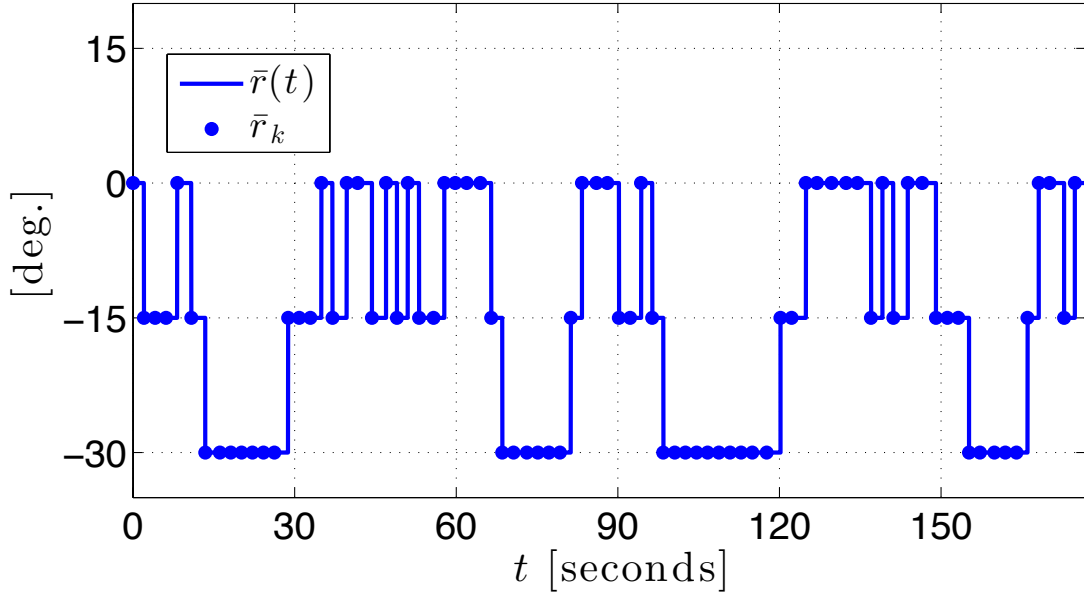


**Figure 2.18:** Critical components of the viewing geometry performance with the stochastic optimal control policy: azimuth  $\vartheta$  and 2-D distance to target  $\rho$ . The mechanical limits of the gimbal,  $\theta_\ell$  and  $\theta_u$ , are indicated by dashed lines in the plot of azimuth angle.

#### 2.4.4 Quantitative Comparison

Quantitative statistics are now presented to solidify the advantages of one technique over the other. Since each approach was tested in flight for a minimum of 15 minutes, statistics are presented for select quantities over a whole 15 minute window for the purpose of an objective comparison. The statistics for critical parameters resulting from both control approaches are presented in





**Figure 2.19:** Roll command sequence under the stochastic optimal control policy  $\mu^*$ .

**Table 2.7:** Statistics over 15 minute window

Statistic	$\gamma^*$	$\mu^*$	Units
$\min \vartheta$	-166.7	-140.3	deg.
$\max \vartheta$	16.63	20.19	deg.
$\text{var } \vartheta$	1026.8	663.3	deg. <sup>2</sup>
$\text{avg } \rho$	150.9	131.8	m
$\text{var } \rho$	2,104	1,372	m <sup>2</sup>
$\text{avg } g(z)$	10.76	7.753	N/A

Table 2.7, where the game theoretic approach is represented by its corresponding control policy  $\gamma^*$  and the stochastic optimal control approach is represented by its corresponding control policy  $\mu^*$ .

The first parameter to consider is the azimuth angle  $\vartheta$ . Both approaches caused the gimbal to saturate and hit each of its mechanical limits of  $-135^\circ$  and  $15^\circ$  at least once, as indicated by the min and max statistics for  $\vartheta$ . However, the azimuth angle never exceeded the FOR limits of  $-152^\circ$  and  $32^\circ$  with the optimal control policy  $\mu^*$ , though it exceeded the lower bound once with the game theoretic control policy  $\gamma^*$ . Moreover, the UAV never lost sight of the target with  $\mu^*$  and lost sight of it just once with  $\gamma^*$ . As noted in the previous section, the azimuth angle varied considerably more with the game theoretic approach than with the stochastic approach, which is indicated by the variance statistic of the said parameter.

Now consider the planar distance to target  $\rho$ . Its average value was nearly 18 [m] less with  $\mu^*$  than with  $\gamma^*$ , and its variance was also significantly less with  $\mu^*$ . Coincidentally, the stochastic optimal control approach also offered a near 35% reduction in the variances of both the azimuth and distance parameters, and finally, it also achieved the lower average viewing geometry cost.

From the quantitative results of Table 2.7, the stochastic optimal control approach proves to be the superior method for the application of vision-based target tracking, as it performed better on all fronts and yielded significantly lower variances for both the azimuth and distance parameters. The most noteworthy

feature of this approach is that it never lost sight of the target even once in its 15 minutes of flight time while maintaining proximity to the target.

Posing the problem of vision-based target tracking as a two-player zero-sum game is a natural way to address the problem when one wishes to ensure robustness to worst-case / evasive target motion. However, for a more casual target vehicle, the precautionary behavior from the behavioral control policy may reduce viewing geometry performance. Another plausible contributor to reduced performance is the fact that the construction of the game theoretic policy assumed first-order roll dynamics, whereas the stochastic approach incorporated a more realistic model of the roll dynamics (Figures 2.6 and 2.7) by treating them as a source of stochasticity. Thus, to improve performance with the game theoretic approach, one could incorporate a second-order model for the roll dynamics of the UAV. Nevertheless, having lost sight of the target only once in 15 minutes, the current game theoretic control policy is still an effective means of vision-based target tracking and potentially much more robust for an evasive target.

## **2.4.5 Wind Considerations**

While there was relatively little wind during flight testing, steady wind generally constitutes a significant portion of a small UAV's airspeed, thereby altering

the aircraft's ground speed significantly. Hence, we performed high fidelity simulations of the existing policies in steady winds to determine their robustness margins to such wind. In particular, the UAV was simulated using the high fidelity flight simulator Aviones while the target trajectory was that encountered during the flight test for the stochastic optimal control policy. Moreover, we conducted multiple 15-minute experiments to test each policy's performance for various constant-speed winds. Since the target's primary direction of travel was west, we simulated steady winds to the east in the flight simulator for the duration of each experiment. Moreover, the UAV primarily experienced a headwind while tracking the target in each experiment. Wind was incorporated into each controller using the approximation described at the end of Section 2.4.1.

The results for the game theoretic control policy and the stochastic optimal control policy are provided in Tables 2.8 and 2.9, respectively, where the middle four columns indicate the absolute wind speed of each scenario. From Table 2.8, the game theoretic policy  $\gamma^*$  exceeded both upper and lower FOR limits in all of the cases that were tested. In addition, the last four quantities in the fourth column (6-m/s scenario) are significantly greater than their counterparts in the third column (4.5-m/s scenario). Hence, even though the increase in wind speed was only 1.5 [m/s], the upper limit of the game theoretic control policy's ability to track in steady wind was crossed as the wind was increased from 4.5 to 6 [m/s].

Thus,  $\gamma^*$  was not tested in 7.5-m/s wind. From Table 2.9, the stochastic optimal control policy  $\mu^*$  fared quite well in the first 3 cases, never losing sight in the first two cases and losing sight of the target only once in the third case, though the target exited the FOR by less than 1 degree, as seen in the first row of the fourth column. In the fourth scenario of 7.5-m/s wind, the last four quantities in the fifth column made significant jumps, which indicated that the boundaries of this policy's capabilities were being crossed, even though the policy lost sight of the target only twice.

Based on these studies, the conservative upper limits for wind speeds that can be tolerated by the policies are 4.5 [m/s] (1/4 of the UAV's airspeed) for the game theoretic case and 6 [m/s] (1/3 of the UAV's airspeed) for the stochastic optimal control case. One would surely improve the UAV's performance in the wind speeds that were tested and perhaps even greater wind speeds by incorporating wind velocity into the models; however, the dimensionality of the problem would grow, thereby presenting considerable computational challenges for basic dynamic programming techniques. Another issue that arises by addressing heavy winds is that the range of target speeds that can be tracked is reduced, as we must have that the combined wind / target speed given by (2.35) must be strictly less than the UAV's airspeed for the tracking problem to be well

**Table 2.8:** Statistics for game theoretic policy in steady wind over 15 minute window

Statistic	3 m/s	4.5 m/s	6 m/s	7.5 m/s	Units
$\min \vartheta$	-165.5	-164.6	-172.3	-	deg.
$\max \vartheta$	61.09	55.16	86.15	-	deg.
$\text{var } \vartheta$	1,378	1,273	2,318	-	deg. <sup>2</sup>
$\text{avg } \rho$	158.5	182.4	217.5	-	m
$\text{var } \rho$	2,967	5,890	9,415	-	m <sup>2</sup>
$\text{avg } g(z)$	12.28	14.81	23.74	-	N/A

**Table 2.9:** Statistics for stochastic optimal control policy in steady wind over 15 minute window

Statistic	3 m/s	4.5 m/s	6 m/s	7.5 m/s	Units
$\min \vartheta$	-132.0	-136.1	-152.6	-131.5	deg.
$\max \vartheta$	8.688	20.22	30.5	45.52	deg.
$\text{var } \vartheta$	634.7	954.6	927.3	1,529	deg. <sup>2</sup>
$\text{avg } \rho$	138.8	140.0	146.3	190.8	m
$\text{var } \rho$	1,692	2,385	2,542	8,181	m <sup>2</sup>
$\text{avg } g(z)$	7.813	8.846	9.244	15.82	N/A

posed. Moreover, there is a tradeoff between the speed of the vehicle that can be tracked and the amount of wind that can be tolerated.

## 2.5 Conclusion

While the existing literature offers methods for target tracking using continuous-time feedback control laws [22, 38, 14, 43, 11, 26, 41] or optimization based methods [2, 36, 32], these individual works make assumptions that simplify the UAV

dynamics, target motion, and/or sensor visibility constraints, thereby hindering the feasibility of a real-world implementation with actual hardware. This chapter has detailed the design of two optimization-based control policies for vision-based target tracking, where strict trajectories must be flown by an underactuated UAV to maintain visibility and proximity to an unpredictable ground target. Both control policies have been flight tested, thereby verifying their robustness to environmental disturbances, unpredictable target motion, and unmodeled dynamics.

The approach based on stochastic optimal control proved most effective through an appropriate design choice of a cost function to facilitate good viewing geometry and also by directly addressing the nontrivial roll dynamics shown in Figure 2.6 and the unpredictable changes in target velocity illustrated in Figure 2.10. This chapter has shown that, with an appropriate choice of cost function, one can use relatively simple kinematic models for motion planning under uncertainty so long as one incorporates sufficient stochasticity into the model.

## Chapter 3

# Optimal UAV Coordination for Vision-based Target Tracking

In this chapter, we study the nature of the optimal coordination strategy for two UAVs performing vision-based target tracking of a moving ground target. More specifically, we investigate the problem of two camera-equipped UAVs tasked with gathering the best *joint* vision-based measurements of the ground target. In this problem formulation, the UAVs travel at a constant altitude and fixed nominal airspeed, which we take to be greater than that of the target. In the previous chapter, we considered only one UAV with sensor visibility constraints and penalized distance in the viewing geometry cost to incorporate the objective of minimizing the geolocation (target localization) error covariance, albeit in a simplified manner. Here we remove the sensor visibility constraint and focus on directly minimizing the geolocation error covariance. Moreover,



one can assume the UAVs are equipped with either an omnidirectional camera or a gimbaled camera with continuous pan-tilt.

## **3.1 Introduction**

In vision-based target tracking, image processing software determines the centroid pixel coordinates of a ground target moving in the image frame. With these pixel coordinates and the intrinsic camera parameters, e.g., focal lengths, skew and image distortion coefficients, etc., one can determine a unit-length line-of-sight vector to the target in the sensor (camera) frame. We assume this vector is provided without noise, as such noise generally constitutes a comparatively small amount of geolocation error [30]. Rather, we focus on the geolocation error arising from the errors in the estimates of the sensor's position and attitude, the latter of which constitutes the dominant source of geolocation error since it is used in a nonlinear fashion to transform the line-of-sight vector from sensor coordinates to topographical coordinates. In addition, since the terrain elevation is used in the flat-Earth approximation to determine the 3-dimensional range to the target by intersecting the UAV's line-of-sight vector (to the target) with the ground plane, errors in the estimate of the terrain elevation also propagate into the geolocation estimation error. Hence, in this chapter, we assume the afore-

mentioned sources of error and provide an analytic expression for an individual UAV's geolocation error covariance (GEC) following the work in [30] and show that the size of the GEC matrix grows nonlinearly as the 3-dimensional distance from the target increases. Thus, for a UAV traveling at a constant altitude, the smallest geolocation error occurs when the UAV is directly above the target, in which case the associated confidence ellipse is circular. However, when the UAV travels away from the target, the GEC grows and the associated confidence ellipse becomes elongated in the viewing direction. Moreover, a UAV would ideally hover above the target, but in general the kinematics of a fixed-wing aircraft prevent this position from being maintained over a period of time.

To mitigate the effects of a single UAV's inability to maintain close proximity to the target, one can employ two UAVs to gather the best *joint* measurements. In this scenario, the objective is to minimize the *fused* geolocation error covariance. Typically the fused geolocation error has the property that it is small when at least one UAV is close to the target and only slightly less when both aircraft are directly above the target. When both UAVs are far from the target relative to their altitudes, the fused geolocation error is greatly reduced when the UAVs have orthogonal viewing angles, though this fused geolocation error is still significantly greater than when at least one UAV is on top of the target. Of course, these configurations are static, and accordingly we formulate an optimal

control problem to determine the nature of the optimal trajectories with the kinematic constraints of fixed-speed aircraft.

### **3.1.1 Related Work**

Much research has proposed coordinated target tracking controllers in a deterministic setting without directly optimizing mission performance with respect to a desired objective function. For two UAVs, a generally accepted practice is to have the UAVs orbit the target at a nominal standoff distance (to remain outside a critical threat range) and maintain an angular separation of  $90^\circ$ . The  $90^\circ$  separation angle minimizes the fused geolocation measurement error for the given standoff distance, as the individual measurement error ellipses are orthogonal [15]. These principles give rise to what is henceforth referred to as cooperative (or coordinated) standoff tracking, which constitutes the majority of the work in the general area of coordinated target tracking. When more than two UAVs are considered, the goal generally becomes having the group achieve a uniform angular separation on a circle centered at the target.

Standoff tracking has been a longstanding goal in the general area of target tracking. In [45], Wheeler et al. employ “Good Helmsman” steering guidance laws developed by Rysdyk (cf. [39]) to achieve coordinated standoff tracking. Frew et al. use Lyapunov guidance vector fields (LGVFs) in [14] to achieve stand-

off tracking with heading-rate control of a single UAV and further use airspeed control to achieve the desired phase separation among multiple UAVs. Summers builds upon the work of Frew in [43] and further assumes that the target and wind speeds are unknown. Using the LGVF approach and adaptive estimates of the combined wind and target velocity, Summers provably achieves the coordinated standoff tracking objective in a decentralized fashion while simultaneously respecting kinematic constraints, e.g., heading rate and airspeed.

Cooperative standoff tracking has received considerable attention in recent years as well. Ma and Hovakimyan provide feedback control laws using turn-rate control to achieve balanced circular formations around a moving ground target under various communication topologies [29]. The uniform angular separation around the target is achieved asymptotically while the standoff distance is uniformly ultimately bounded. In [33], Oh et al. propose a tangent vector field guidance strategy for coordinated standoff tracking with multiple UAVs wherein they employ sliding mode control with adaptive terms in order to mitigate the effects of disturbances and modeling uncertainty in the turn-rate dynamics. With estimates of the target state from a decentralized information filter, the UAVs achieve a desired angular separation in a decentralized fashion under various information / communication architectures by using either airspeed control or orbit radius change.

Since multiple fixed-speed aircraft cannot maintain a uniform angular spread at a fixed distance from a constant velocity target, some works have explored the notion of spreading agents uniformly in time along a periodic trajectory at a fixed distance from the moving target. In [21], Kingston developed a coordinated controller to achieve a time-splay configuration using sliding mode control. Peterson and Paley achieved the same goal in [35] by assuming an all-to-all communication topology and developing theoretically justified decentralized control laws that achieve time splay configurations for not only a constant velocity target, but also a constant-speed target turning at a constant rate. Bounds on the maximum turn rate for such controllers are also provided.

With the exception of [35], all of the preceding works utilize either nonlinear, adaptive, or LGVF control techniques and either assume a constant velocity for the target or make no explicit considerations of maximum control input (either turn-rate or bank angle) for the UAV. However, a number of approaches have employed stochastic optimal control to mitigate the effects of stochastic target motion and optimally estimate the target state while respecting a maximum turn rate. Miller et al. pose the target tracking problem as a partially observable Markov decision process (POMDP) in [32] and present a new approximate solution, as nontrivial POMDP problems are intractable to solve exactly [44]. In particular, the authors consider the case of multiple UAVs tracking multi-

ple targets and employ heuristics in evaluating the  $Q$ -value (expected cost for a particular control action) to overcome the limitations of short planning horizons in the presence of occlusions. Stachura et al. [42] studied the problem of two variable-airspeed UAVs tracking a stochastic ground target using bearing-only sensors in the presence of packet losses when communicating with the base station, where target state estimation takes place. The solution involved an online receding horizon controller that maximized the expected information over a short planning horizon, showing that one UAV will act as a relay to the base station when the target is far from the base. In [9], Ding et al. studied the problem of optimally coordinating two camera-equipped Dubins vehicles with bang-off-bang turn-rate control to maximize the geolocation information of a stochastic ground target over a short planning horizon. The results showed that a  $90^\circ$  separation in the viewing angle was essential in the case of terrestrial pursuit vehicles and less pronounced when the pursuit vehicles were airborne.

We emphasize the fact that the preceding optimal control approaches illustrate a trend among optimization-based coordination strategies. Namely, shorter planning horizons are often considered so that one can employ online receding-horizon control strategies and avoid the challenges associated with dynamic programming. While this is easily justified from a pragmatic standpoint, short horizons are not adequate for the cost function considered here.

While the literature review presented here is by no means an exhaustive list of coordinated target tracking techniques, it is certainly representative of the state of the art. However, in all of the preceding works, at least one or more assumptions are made that impose severe practical limitations. Namely, the works mentioned thus far assume at least one of the following:

1. Coordinated circular trajectories are optimal, namely those resulting from standoff tracking.
2. Short/greedy planning horizons are adequate for optimal tracking.
3. Airspeed can be changed quickly and reliably over a significant range.
4. Input dynamics are first order, which implies that roll dynamics have been ignored.
5. Target motion is predictable.

The work presented in this chapter address the first three of the preceding points in an ideal, deterministic setting in order to study the nature of the optimal trajectories without the effects of stochasticity and higher order nonlinear dynamics. In particular, we consider two UAVs modeled as Dubins vehicles tracking a constant-velocity target with no restrictions on the motion of the agents other than kinematics. The stochastic optimal control problem involving fourth-order

stochastic kinematic models similar to those of Chapter 2 will be covered in the next chapter, as the dimensionality of the problem renders grid-based dynamic programming solutions, such as the ones considered here and in the previous chapter, infeasible. Consequently, the next chapter will address all five of the listed assumptions by providing a regression-based dynamic programming solution to a more practical problem formation involving higher order stochastic dynamics with explicit input constraints, thus yielding optimal coordination under more realistic conditions. The present work is rather focused on deriving an analytical expression for the geolocation error covariance and studying the behavior of two UAVs tracking a moving ground target in an ideal setting.

### **3.1.2 Organization of Chapter**

The remainder of this chapter is organized as follows. Section 3.2 provides a discussion of the system dynamics, the derivation of the geolocation error covariance, and finally a description of the dynamic programming solution. Section 3.3 presents simulations results for multiple scenarios and characterizes the nature of the optimal solution. It concludes by comparing the performance of the optimal controller to that of a coordinated standoff tracking controller. Finally, Section 3.4 summarizes the results.



## 3.2 Problem Formulation

Consider a group of two UAVs tasked with gathering the best joint vision-based measurements of a moving ground target. The UAVs fly at a fixed forward speed while maintaining a constant altitude. The target vehicle moves on the ground, and has a forward speed that can be significantly slower than that of the UAVs. Each UAV makes measurements of the target using a gimbaled video camera. The main objective is to optimize the coordination of the UAVs with respect to the fused geolocation (target localization) error covariance. In this section, we first describe the Dubins vehicle model for the UAVs, as well as the overall state space. We then discuss the measurement model and the resulting geolocation error covariance.

### 3.2.1 Vehicle Modeling

The Dubins vehicle is a planar vehicle that moves forward at a fixed speed and has a bounded turning radius and is commonly used to provide a simple model for UAVs flying at a fixed altitude. We assume that UAV  $j$ , where  $j \in \{1, 2\}$ , flies at a constant speed  $s_j$  and at a fixed altitude  $h_j$ , and it has a bounded turning rate  $u_j$  with maximum absolute upper bound  $\bar{u} \in \mathbb{R}_{>0}$ , which is the same for both agents. Accordingly,  $u \in \mathcal{U} \subseteq [-\bar{u}, \bar{u}] \times [-\bar{u}, \bar{u}]$ . Let  $(x_j, y_j) \in \mathbb{R}^2$  denote

agent  $j$ 's planar position while  $\psi_j \in \mathbb{S}^1$  denotes its heading. We define UAV  $j$ 's pose as  $\mathbf{p}_j := (x_j, y_j, \psi_j)$ . Then the kinematics of UAV  $j$  are given by

$$\frac{d\mathbf{p}_j}{d\tau} = \begin{pmatrix} s_j \cos \psi_j \\ s_j \sin \psi_j \\ u_j \end{pmatrix}. \quad (3.1)$$

The optimization will again be done in discrete time, and hence we apply a  $T_s$ -second zero-order hold (ZOH) to each control input  $u_j$ . For  $u_j \neq 0$ , the equations of motion become

$$\mathbf{p}_j^+ = \begin{pmatrix} \frac{s_j}{u_j}(\sin(u_j T_s + \psi_j) - \sin(\psi_j)) \\ \frac{s_j}{u_j}(\cos(\psi_j) - \cos(u_j T_s + \psi_j)) \\ u_j T_s \end{pmatrix} + \mathbf{p}_j \quad (3.2)$$

whereas for  $u_j = 0$ , we have

$$\mathbf{p}_j^+ = \begin{pmatrix} s_j T_s \cos(\psi_j) \\ s_j T_s \sin(\psi_j) \\ 0 \end{pmatrix} + \mathbf{p}_j. \quad (3.3)$$

In general we assume the target is a nonholonomic vehicle with planar position  $(x_g, y_g) \in \mathbb{R}^2$  and heading  $\psi_g \in \mathbb{S}^1$ . Hence, we define the target's pose as  $\mathbf{p}_g := (x_g, y_g, \psi_g)$ . In the present chapter, the target is assumed to be traveling

at a constant velocity with speed  $v$ , and hence, without loss of generality, its dynamics are simply

$$\frac{d\mathbf{p}_g}{d\tau} = \begin{pmatrix} v \\ 0 \\ 0 \end{pmatrix}, \quad (3.4)$$

with  $y_g(0) = 0$  and  $\psi_g(0) = 0$ .

The equivalent discrete-time dynamics with a  $T_s$ -second ZOH are simply

$$\mathbf{p}_g^+ = \begin{pmatrix} vT_s \\ 0 \\ 0 \end{pmatrix} + \mathbf{p}_g. \quad (3.5)$$

We denote the overall state space by  $\mathcal{Z} \subseteq \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{S}^1$ , and its elements are defined as

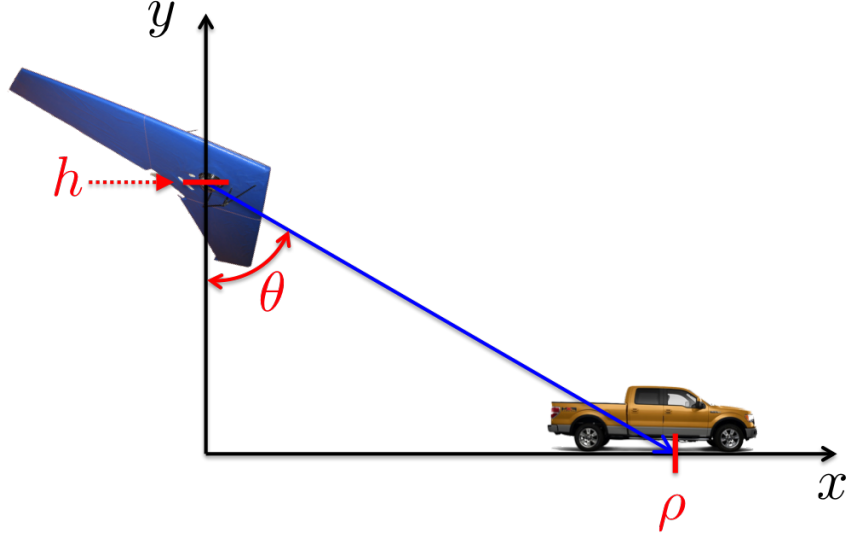
$$z := (\mathbf{p}_1 - \mathbf{p}_g, \mathbf{p}_2 - \mathbf{p}_g). \quad (3.6)$$

The overall dynamics are thus given by

$$z^+ = f(z, u) := \begin{pmatrix} \mathbf{p}_1^+ - \mathbf{p}_g^+ \\ \mathbf{p}_2^+ - \mathbf{p}_g^+ \end{pmatrix}. \quad (3.7)$$

### 3.2.2 Geolocation Error Covariance

Each UAV has a video sensor that makes image-plane measurements of the target. The geolocation error arises from the errors in the estimates of sensor



**Figure 3.1:** Illustration of attitude error propagation. In this plot, the known UAV position is  $(0, h)$  while the target's true position to be estimated is  $(\rho, 0)$ . Suppose the measured sensor attitude  $\theta$  is corrupted by zero-mean Gaussian noise  $\tilde{\theta}$  with variance  $\sigma_{\theta}^2$ , leaving us with an estimate of the sensor attitude given by  $\hat{\theta} = \theta + \tilde{\theta}$ . Then one can take  $\hat{\rho} = h \tan(\hat{\theta})$ , and show that linearizing about  $\tilde{\theta} = 0$  yields  $\text{Var}(\hat{\rho}) = (h \sec^2 \theta)^2 \sigma_{\theta}^2$ , which is very nonlinear for  $\rho \gg h$ .

attitude, sensor position, and terrain height. As noted in [30], the dominant source of geolocation error stems from the error in the sensor attitude matrix that relates the line-of-sight vector in the sensor frame (centered at the UAV position) to that in the topographic coordinate frame [30]. We provide an intuitive one dimensional example in Figure 3.1 and note that the techniques used to derive the geolocation error in 3 dimensions are similar.

We relate each UAV's image-plane measurements to topographical coordinates following the work of Mallick [30] and include all three sources of error in the derivation of the geolocation error covariance. This work is presented not

only for completeness, but also for improved clarification of the exposition. In following Mallick’s work, we also use boldface notation in this section and note that such quantities should be distinguished from their non-boldface counterparts elsewhere.

Video tracking uses two main coordinate frames: the topographic coordinate frame ( $T$  frame) and the sensor coordinate frame ( $S$  frame). The  $T$  frame is the primary coordinate frame, and its  $x$ ,  $y$ , and  $z$  axes are along the East, North, and upward directions, respectively. The  $S$  frame is the secondary frame and has its origin at the optical center of the camera. In the  $S$  frame, the  $z$ -axis is along the optical axis and is pointed in the general downward direction toward the Earth. Also, superscript “ $S$ ” denotes a quantity in the sensor coordinate frame; however, we omit the superscript “ $T$ ” on vector quantities in the topographic coordinate frame, as this is the assumed primary coordinate frame.

To perform the coordinate transformation from the  $T$  frame to the  $S$  frame, one uses the orthogonal attitude matrix  $T_T^S(\boldsymbol{\theta})$ , which is a nonlinear function of the 3-2-1 Euler-angle sequence of yaw, pitch, and roll denoted by  $\boldsymbol{\theta} \in \mathbb{R}^3$ . We note that the Euler angles are those of the camera sensor and not the aircraft. Furthermore, the inverse transformation is denoted by  $T_S^T(\boldsymbol{\theta})$  and is equal to the

following matrix:

$$\begin{bmatrix} c\theta_2 s\theta_1 & c\theta_3 c\theta_1 + s\theta_3 s\theta_2 s\theta_1 & -s\theta_3 c\theta_1 + c\theta_3 s\theta_2 s\theta_1 \\ c\theta_2 c\theta_1 & -c\theta_3 s\theta_1 + s\theta_3 s\theta_2 c\theta_1 & s\theta_3 s\theta_1 + c\theta_3 s\theta_2 c\theta_1 \\ s\theta_2 & -s\theta_3 c\theta_2 & -c\theta_3 c\theta_2 \end{bmatrix}, \quad (3.8)$$

where  $c\theta_i = \cos \theta_i$  and  $s\theta_i = \sin \theta_i$ . Finally, we assume that image tracking software is available to control the camera's gimbal platform and keep the target in the camera's field of view, as in [18]. Moreover, image tracking software determines and reports the Euler-angle sequence  $\boldsymbol{\theta}$ .

Let  $\mathbf{s} \in \mathbb{R}^3$  and  $\mathbf{o} \in \mathbb{R}^3$  represent the true sensor and ground target positions, respectively, measured from the  $T$ -frame origin. Throughout this monograph, we take the sensor position to be equal to the UAV position. Moreover, for a UAV with planar position  $(x, y)$  and altitude  $h$  in the  $T$  frame, we take  $\mathbf{s} = (x, y, h)$ . The relationship between the object and image point is defined through the attitude matrix,

$$\mathbf{r}^S := \mathbf{o}^S - \mathbf{s}^S = T_T^S(\boldsymbol{\theta})(\mathbf{o} - \mathbf{s}), \quad (3.9)$$

and we let  $d := \|\mathbf{r}^S\|_2$  and  $\mathbf{u}^S := \mathbf{r}^S/d$ . Here,  $d$  is the 3-dimensional distance from the  $S$ -frame origin to the target position, and  $\mathbf{u}^S \in \mathbb{R}^3$  is the unit vector along the camera's line-of-sight.

The relation between the sensor and target positions can be estimated as

$$\hat{\mathbf{o}} = \hat{\mathbf{s}} + d\hat{T}_S^T(\hat{\boldsymbol{\theta}})\mathbf{u}^S. \quad (3.10)$$

The quantity  $\hat{d}$  is estimated using the flat-Earth approximation. In practice,  $\mathbf{u}^S$  can be determined from the observed pixel coordinates and camera parameters and is not treated as a source of error in this work, though it can certainly be incorporated into the model presented here.

The quality of a particular measurement depends on the location of the UAV with respect to the target. When the UAV is far from the target, relative to its height, the estimation error covariance is intuitively elongated and we now quantify this covariance for the video measurements according to [30]. The main, uncorrelated sources of error arise from the estimates for sensor position ( $\hat{\mathbf{s}}$ ), sensor attitude angles ( $\hat{\boldsymbol{\theta}}$ ), and terrain height ( $\hat{h}_0$ ). Hence, the error models are:

$$\begin{aligned}\hat{\mathbf{s}} &= \mathbf{s} + \tilde{\mathbf{s}}, \quad \tilde{\mathbf{s}} \sim \mathcal{N}(0, R_{\tilde{\mathbf{s}}}), \\ \hat{\boldsymbol{\theta}} &= \boldsymbol{\theta} + \tilde{\boldsymbol{\theta}}, \quad \tilde{\boldsymbol{\theta}} \sim \mathcal{N}(0, R_{\tilde{\boldsymbol{\theta}}}), \\ \hat{h} &= h_0 + \tilde{h}_0, \quad \tilde{h}_0 \sim \mathcal{N}(0, \sigma_h^2).\end{aligned}\tag{3.11}$$

Given the estimated geolocation in (3.10), the corresponding geolocation error can be written as

$$\tilde{\mathbf{o}} := \hat{\mathbf{o}} - \mathbf{o} \approx \tilde{\mathbf{s}} + \hat{\mathbf{u}}\tilde{d} + \hat{d}\tilde{\mathbf{u}},\tag{3.12}$$

where we have ignored the higher order term  $\tilde{d}\tilde{\mathbf{u}}$ . Also,  $\hat{\mathbf{u}} = T_S^T(\hat{\boldsymbol{\theta}})\mathbf{u}^S$ . We now seek to transform this expression into one that is linear in the random variables /

vectors given in (3.11) in order for the geolocation error to be fully characterized by a (state-dependent) normal distribution. To this end, we begin by linearizing the estimated attitude matrix  $T_S^T(\hat{\boldsymbol{\theta}})$  about the operating point  $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$  to obtain

$$\hat{\mathbf{u}} = T_S^T(\boldsymbol{\theta} + \tilde{\boldsymbol{\theta}})\mathbf{u}^S \approx T_S^T(\boldsymbol{\theta})\mathbf{u}^S + A(\mathbf{u}^S, \boldsymbol{\theta})\tilde{\boldsymbol{\theta}}, \quad (3.13)$$

where

$$A(\mathbf{u}^S, \boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} (T_S^T(\boldsymbol{\theta})\mathbf{u}^S). \quad (3.14)$$

Thus, in (3.12), we have

$$\tilde{\mathbf{u}} = \hat{\mathbf{u}} - \mathbf{u} = A(\mathbf{u}^S, \boldsymbol{\theta})\tilde{\boldsymbol{\theta}} \quad (3.15)$$

Next we need to define a relationship for the term  $\tilde{d}$  in (3.12). Starting from the definition for the estimated object position (3.10), and using the flat-Earth approximation,

$$\hat{d} = \frac{\hat{h}_0 - \hat{s}_3}{\hat{u}_3}. \quad (3.16)$$

By substituting  $h_0 + \tilde{h}_0$  for  $\hat{h}_0$ ,  $s_3 + \tilde{s}_3$  for  $\hat{s}_3$ , and  $u_3 + \tilde{u}_3$  for  $\hat{u}_3$ , we obtain

$$\tilde{d} = \frac{\tilde{h}_0 - \tilde{s}_3}{u_3 + \tilde{u}_3} - \left( \frac{h_0 - s_3}{u_3} \right) \left( \frac{\tilde{u}_3}{u_3 + \tilde{u}_3} \right). \quad (3.17)$$

Treating  $\tilde{d}$  as a function of  $\boldsymbol{\zeta} := (\tilde{h}_0, \tilde{s}_3, \tilde{u}_3)$ , and linearizing about  $\boldsymbol{\zeta} \equiv 0$  yields

$$\tilde{d} = \frac{\tilde{h}_0 - \tilde{s}_3}{u_3} - \frac{h_0 - s_3}{u_3^2} \tilde{u}_3. \quad (3.18)$$



By substituting (3.15) and (3.18) into (3.12), we obtain

$$\tilde{\mathbf{o}} = B\tilde{\mathbf{s}} + \frac{\mathbf{u}}{u_3}\tilde{h}_0 + dBA(\mathbf{u}^S, \boldsymbol{\theta})\tilde{\boldsymbol{\theta}}, \quad (3.19)$$

where

$$B := I_{3 \times 3} - \mathbf{u} \begin{bmatrix} 0 & 0 & 1/u_3 \end{bmatrix}. \quad (3.20)$$

By computing the covariance of (3.19), we arrive at the geolocation error covariance (GEC),

$$P_{\tilde{\mathbf{o}}} = BR_{\tilde{\mathbf{s}}}B^\top + \frac{1}{u_3^2}\mathbf{u}\mathbf{u}^\top\sigma_h^2 + \dots \quad (3.21)$$

$$+ d^2BA(\mathbf{u}^S, \boldsymbol{\theta})R_{\tilde{\boldsymbol{\theta}}}A^\top(\mathbf{u}^S, \boldsymbol{\theta})B^\top, \quad (3.22)$$

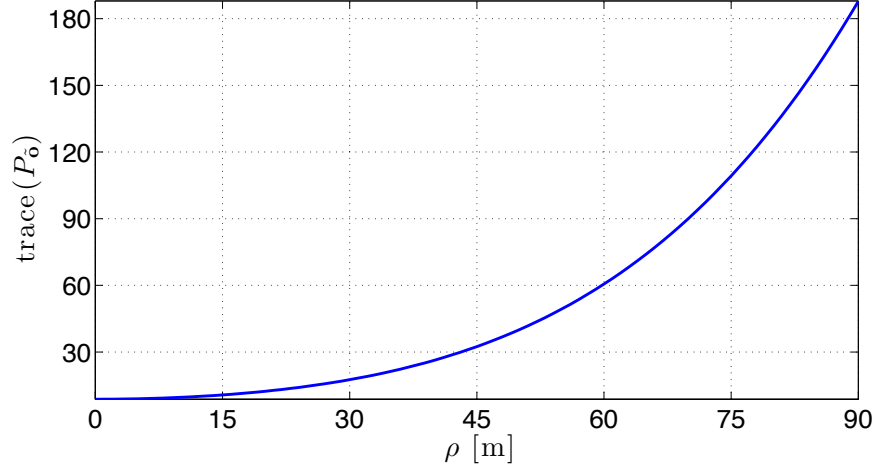
where  $R_{\tilde{\mathbf{s}}}$ ,  $\sigma_h^2$ , and  $R_{\tilde{\boldsymbol{\theta}}}$  are given in (3.11).

The dominant source of error arises from the uncertainty in the sensor attitude angles. Moreover, we let  $R_{\tilde{\mathbf{s}}} \equiv 0$  and  $\sigma_h^2 = 0$ , in which case the GEC from (3.22) simplifies to

$$P_{\tilde{\mathbf{o}}} = d^2BA(\mathbf{u}^S, \boldsymbol{\theta})R_{\tilde{\boldsymbol{\theta}}}A^\top(\mathbf{u}^S, \boldsymbol{\theta})B^\top. \quad (3.23)$$

Finally, because tracking will be done in the ground plane, only the  $2 \times 2$  upper-left submatrix of  $P_{\tilde{\mathbf{o}}}$  is relevant, and for UAV  $j$ , we take

$$P_j := \begin{bmatrix} P_{\tilde{\mathbf{o}}:1,1} & P_{\tilde{\mathbf{o}}:1,2} \\ P_{\tilde{\mathbf{o}}:2,1} & P_{\tilde{\mathbf{o}}:2,2} \end{bmatrix}. \quad (3.24)$$



**Figure 3.2:** Trace of the GEC as a function of planar distance  $\rho$  for an individual UAV with the target fixed at the origin ( $\mathbf{o} = 0$ ) in the  $T$ -frame. Moreover, the UAV's position in the  $T$  frame is  $\mathbf{s} = (\rho, 0, 40)$ , which yields  $P_{\hat{\mathbf{o}}:1,2} = P_{\hat{\mathbf{o}}:2,1} = 0$ . Hence the plot depicts the nonlinear growth of the total variance.

Without loss of generality we assume  $R_{\hat{\boldsymbol{\theta}}} = 9I_{3 \times 3}$ , which has units of degrees squared. The standard deviation of 3 degrees for each Euler angle is in the middle of the range considered in [30]. The nonlinear growth of the total uncertainty for a single UAV is depicted in Figure 3.2.

With the UAVs collecting independent measurements of the target, the fused GEC  $\mathcal{P}$  can be computed according to the following relationship

$$\mathcal{P}^{-1} = \sum_j P_j^{-1}. \quad (3.25)$$

The nature of the error covariances, both individual and fused, is illustrated in Figure 3.3. Note that the fused covariance is determined by three degrees of freedom, namely the planar distances from the target denoted by  $\rho_j$  for  $j \in \{1, 2\}$

and the separation angle  $\gamma$ . By denoting UAV  $j$ 's planar position *relative* to the target as  $\mathbf{r}_j \in \mathbb{R}^2$ , i.e.,

$$\mathbf{r}_j := (x_j - x_g, y_j - y_g), \quad (3.26)$$

the separation angle is given implicitly as

$$\mathbf{r}_1^\top \mathbf{r}_2 = \rho_1 \rho_2 \cos \gamma, \quad (3.27)$$

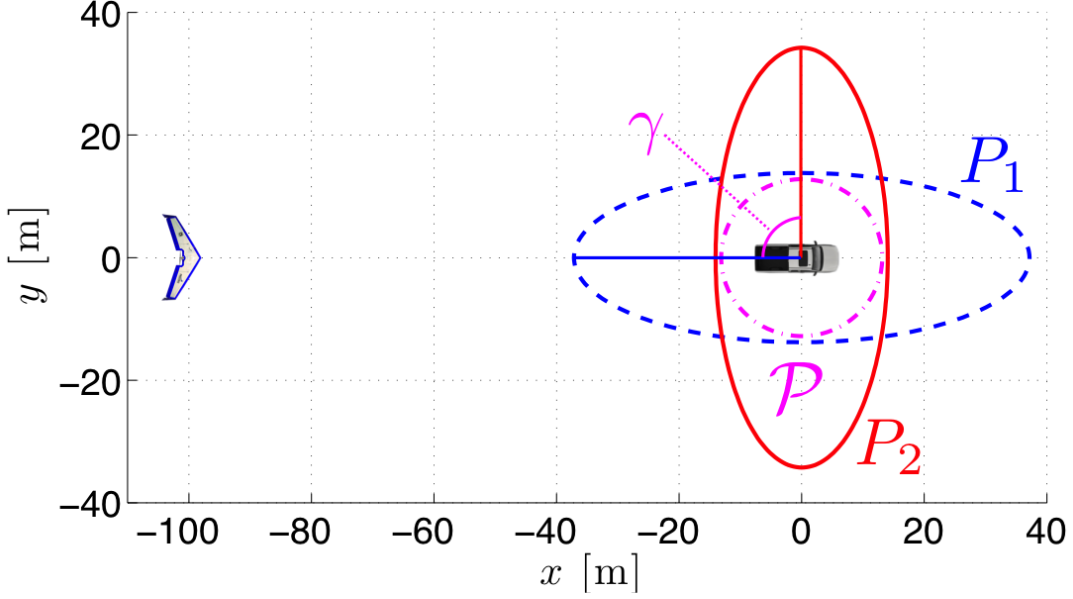
where  $\rho_j = \|\mathbf{r}_j\|_2$ . We take  $\gamma$  to be zero when either  $\rho_1 = 0$  or  $\rho_2 = 0$ . Note that  $z = (\mathbf{r}_1, \psi_1, \mathbf{r}_2, \psi_2)$ , and the line-of-sight vector from UAV  $j$  to the target is  $\mathbf{r}_j = (-\mathbf{r}_j, -h_j)$ , where we have assumed the terrain height is  $h_0 = 0$  [m].

When performing a minimization involving a matrix, a common practice is to instead minimize a corresponding real-valued scalar objective function. While the specific choice of matrix scalarization is debatable, we opt for minimizing the trace of the GEC in this work, as it is equivalent to minimizing the average variance in all directions [36]. One can certainly employ alternative matrix scalarizations as desired. Thus, utilizing the state space definition of Section 3.2.1, we take the objective function of the optimal control problem to be

$$g(z) := \text{trace}(\mathcal{P}), \quad (3.28)$$

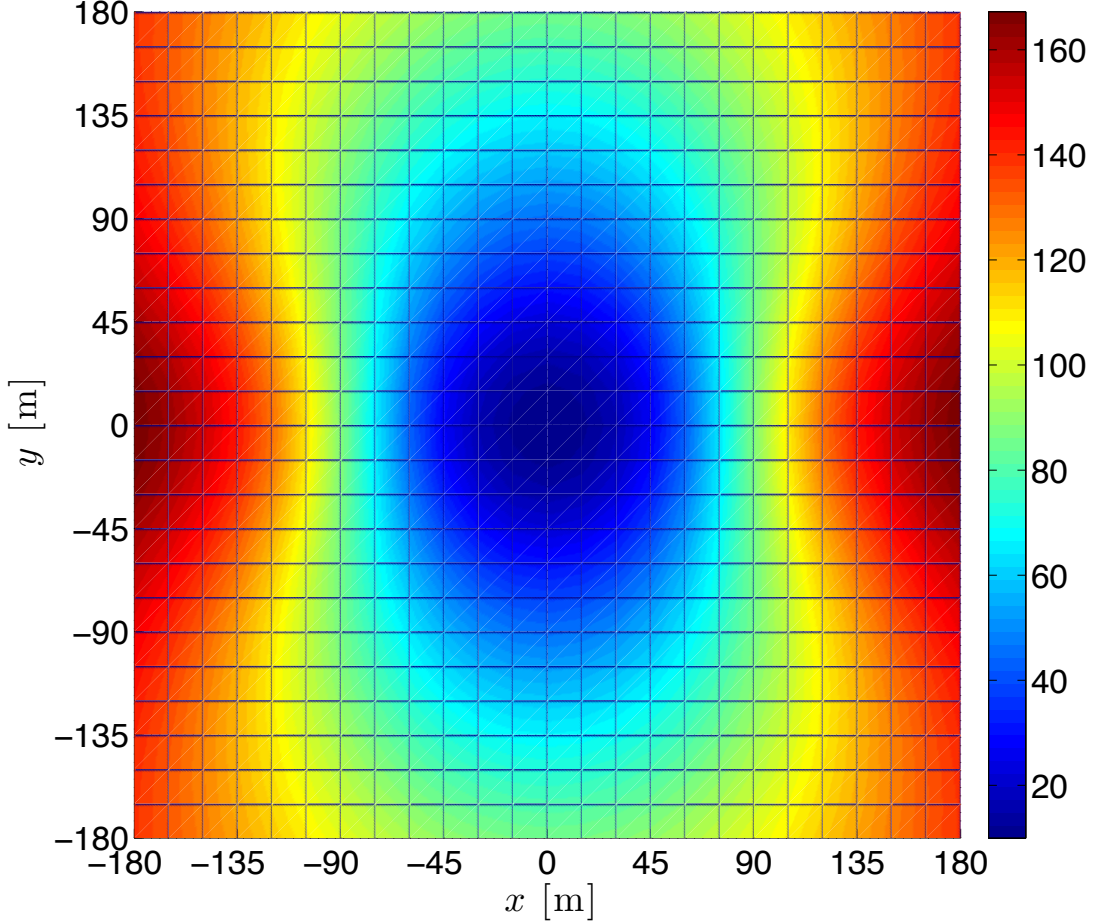
which has units of meters squared.

The nature of this cost function is illustrated in Figure 3.4 for two UAVs. Note that if the second UAV's position is on the  $x$ -axis, then the UAVs are



**Figure 3.3:** Individual error ellipses  $P_1$  and  $P_2$  corresponding to the vision measurements from the blue and red UAVs having  $(x, y, z)$  coordinates (in meters) of  $(-100, 0, 40)$  and  $(0, 100, 45)$  respectively, where the latter UAV is not shown. Also depicted is the error ellipse  $\mathcal{P}$  corresponding to the combination (fusion) of the measurements obtained from both UAVs, and the separation angle  $\gamma$ .

collinear, which entails that the major axes of their error ellipses are perfectly aligned. If however, its position is on the  $y$ -axis, then the UAVs have orthogonal viewing angles. Thus, one can see that the UAVs certainly benefit from having orthogonal viewing angles. One should also note how the trace of the first UAV's *individual* GEC is above  $180 \text{ [m}^2\text{]}$  per Figure 3.2 while that of the *fused* covariance is drastically lower for certain positions of the second UAV. In particular, if the second UAV's  $(x, y)$ -position is  $(0, 90)$ , such that the UAVs have orthogonal viewing angles, then  $\text{trace}(\mathcal{P}) \approx 46 \text{ [m}^2\text{]}$ . If, however, the second UAV is on



**Figure 3.4:** Cost function  $g(z) = \text{trace}(\mathcal{P})$  with the target located at the origin and the first UAV located at three dimensional position  $(x, y, z) = (90, 0, 40)$ . Note that the first UAV's individual GEC is the same as that in Figure 3.2 for  $\rho = 90$  [m]. The second UAV has an altitude of 45 [m], and the  $(x, y)$  coordinates in the plot represent its planar position.

top of the target, then  $\text{trace}(\mathcal{P}) \approx 10$  [m<sup>2</sup>]. Thus, an effective coordination strategy would be to have at least one UAV close to the target (if possible), as a UAV's individual GEC is smallest in this setting and will dominate the cost through (3.25). Hence, we now formulate the optimal control problem

whose solution will determine what is indeed possible when vehicle dynamics are involved.

### 3.2.3 Problem Statement and Solution

The optimal control problem is to determine the optimal feedback control policy  $\mu_k^* : \mathcal{Z} \rightarrow \mathcal{U}$ ,  $k \in \{0, 1, \dots, K-1\}$ , that minimizes

$$J(z_0) = \sum_{k=0}^K g(z_k), \quad \forall z_0 \in \mathcal{Z}, \quad (3.29)$$

where  $z_k = z(kT_s)$ ,  $K \in \mathbb{N}$ ,  $g(\cdot)$  is given by (3.28), and  $z$  evolves deterministically according to  $f(z, u)$  defined by (3.7) under the feedback law  $u_k = \mu_k^*(z)$ . To solve this problem, we employ *value iteration* in a deterministic setting.

In the deterministic case, the *value function*, or *cost-to-go* from state  $z \in \mathcal{Z}$  at time  $k \in \{0, 1, \dots, K-1\}$ , is defined as

$$V_k(z) := g(z) + \min_{u_k, u_{k+1}, \dots, u_{K-1}} \sum_{\ell=k+1}^K g(z_\ell). \quad (3.30)$$

For  $k = K$ , one takes  $V_K(z) = g(z)$ , and the cost-to-go for  $k \in \{0, 1, \dots, K-1\}$  is computed (offline) in reverse chronological order according to the recursion

$$V_k(z) = \min_{u \in \mathcal{U}} \left( g(z) + V_{k+1}(f(z, u)) \right), \quad (3.31)$$

which holds due to Bellman's principle of optimality (see [25], Chapter 6). As the minimization is performed, the optimal control policy is also formed as

$$\mu_k^*(z) = \arg \min_{u \in \mathcal{U}} \left( g(z) + V_{k+1}(f(z, u)) \right). \quad (3.32)$$

Moreover, if one knows the optimal policy from time  $k + 1$  onward, and hence the value function at time  $k + 1$ , one can extend it to the previous time  $k$  by performing the minimization in (3.31). Performing the sequence of computations in (3.31) for  $k \in \{0, 1, \dots, K - 1\}$  ultimately yields  $J^*(z) = V_0(z)$ ,  $\forall z \in \mathcal{Z}$ , where  $J^*(z)$  is the minimum value of (3.29) under the feedback law (3.32).

Time-optimal Dubins vehicle trajectories comprise three motion primitives, namely turning left at the maximum allowable turn rate ( $u_j = -\bar{u}$ ), going straight ( $u = 0$ ), or turning right at the maximum allowable turn rate ( $u_j = \bar{u}$ ). Accordingly, we assume  $u \in \mathcal{U} := \{0, \pm\bar{u}\} \times \{0, \pm\bar{u}\}$ . To limit the computation of the value function to a finite number of points, one works with a finite subset  $Z$  of the state space  $\mathcal{Z}$  having  $N_s$  distinct elements and the quantization function  $q(z, Z)$  given by (2.2). Based on these, one approximates the computation of the value function and optimal policy by

$$V_k(z) \approx g(z) + \min_{u \in \mathcal{U}} V_{k+1}(q(f(z, u), Z)) \quad (3.33)$$

$$\mu_k^*(z) = \arg \min_{u \in \mathcal{U}} \left[ g(z) + V_{k+1}(q(f(z, u), Z)) \right], \quad (3.34)$$

which only require the evaluation of the value function over the finite set  $Z$ .

In practice, to lookup the optimal command  $u_k$  for a state  $z \in \mathcal{Z} \setminus Z$ , one takes  $u_k = \mu_k^*(q(z, Z))$ .

### 3.3 Simulation Results

We now present and study the simulation results for multiple dynamic optimizations with the parameters of Table 3.1 in order to understand the nature of the optimal UAV trajectories for various target speeds and at different altitudes. In particular, we first demonstrate the effect altitude has on the importance of maintaining a  $90^\circ$  separation angle by presenting the results for the baseline altitudes presented in Table 3.1 as well as the results obtained with each of the altitudes doubled. Secondly, we demonstrate results for target speeds that are 30%,  $33\frac{1}{3}\%$ , and 70% of the UAV's airspeed, which we take to be equivalent to its ground speed in an ideal setting of zero wind. The lower two target speeds are similar and demonstrate how the nature of the optimal trajectories switches from circular orbits at lower speeds to sinusoids or "S" turns at higher speeds. The fastest target speed illustrates the nature of the optimal trajectories as the target speed approaches that of the UAV. We conclude the study with a com-



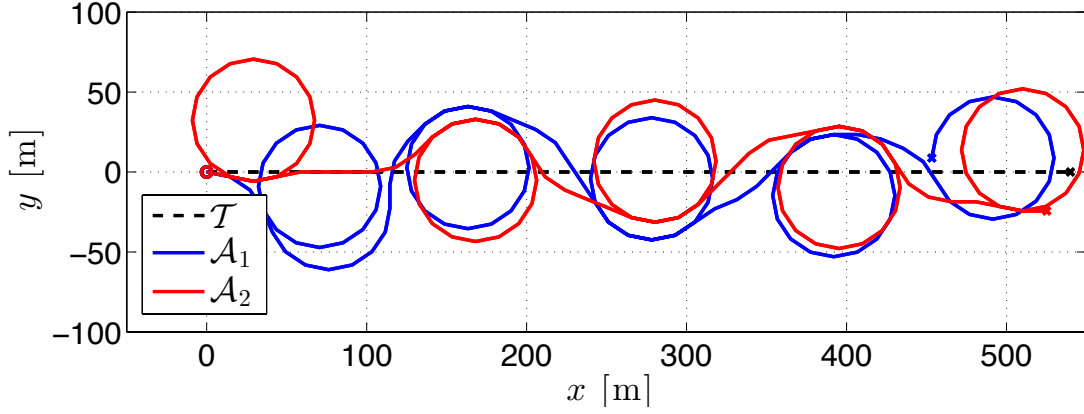
Symbol	Description	Value	Units
$(h_1, h_2)$	UAV altitudes	$(40, 45)$	m
$(s_1, s_2)$	UAV airspeeds	$(15, 15)$	m/s
$v$	target speeds	$\{4.5, 5, 10.5\}$	m/s
$K$	planning horizon	60	-
$T_s$	sampling time	1	s
$R_{\hat{\theta}}$	sensor attitude cov.	$9I_{3 \times 3}$	deg <sup>2</sup>
$\bar{u}$	max turn rate	22.5	deg./s
$X$	relative positions	$\{-100, -95, \dots, 100\}$	m
$\Psi$	relative headings	$\{0, 22.5, \dots, 337.5\}$	deg.
$Z$	discrete state space	$X^2 \times \Psi \times X^2 \times \Psi$	-

**Table 3.1:** Simulation Parameters

parison of the performance of the optimal controller with that generated by the splay-state controller of Kingston [21].

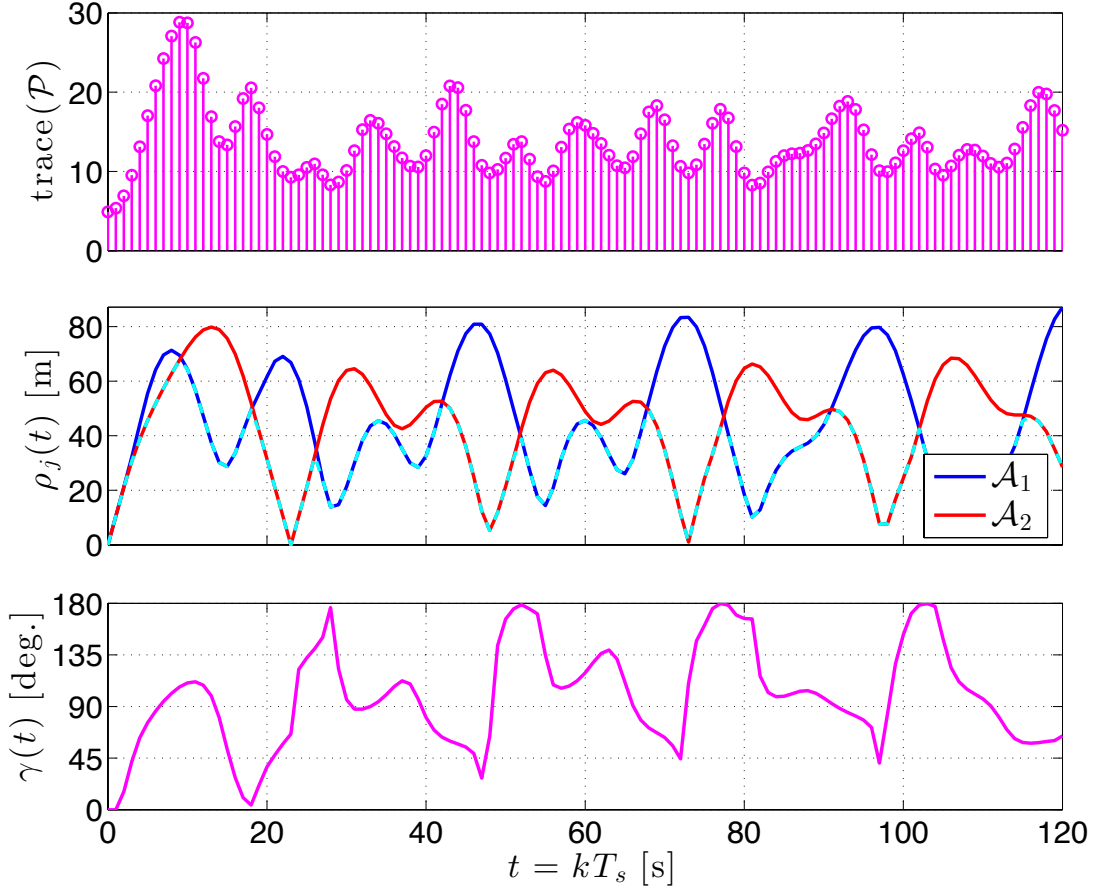
### 3.3.1 Study of Simulation Results

We begin by examining the optimal trajectories for the slowest target speed of  $v = 4.5$  [m/s]. The target begins at the origin of the  $xy$  plane and the initial condition for the relative coordinates is  $z_0 \equiv 0$ . This particular initial configuration will be the same for all simulations presented in this section. The UAV trajectories for the present scenario are plotted in Figure 3.5 while corresponding performance metrics are provided in Figure 3.6. From Figure 3.5, we observe the UAVs performing circular orbits in the same direction that also pass through the target's paths. By inspecting the distance plot of Figure 3.6, one can observe



**Figure 3.5:** UAV trajectories over a two-minute window for a target speed of  $v = 4.5$  [m/s] and an initial condition of  $z_0 = 0$ . The UAV trajectories comprise quasi-periodic circular orbits of alternating direction that pass over or near to the target. The passes over the target are coordinated to be out of phase, as indicated by the distance curves of Figure 3.6.

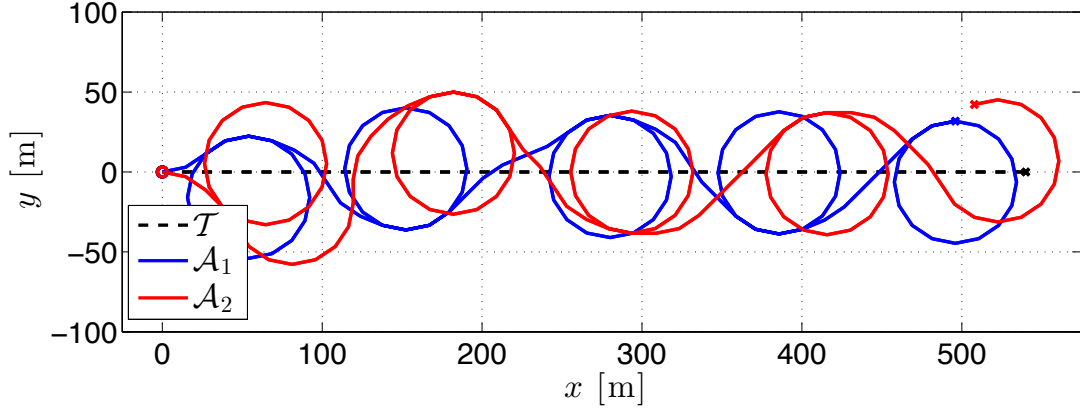
that the UAV's pass over (or close) to the target along quasi-periodic trajectories. Furthermore, one should note that when the second UAV ( $\mathcal{A}_2$ ) achieves its minimum relative distance, e.g., at times  $t \in \{23, 48, 73, 97\}$ , the first UAV ( $\mathcal{A}_1$ ) is near its peak distance. Moreover, *distance coordination* is evident. During the initial transient phase both UAVs become relatively far from the target and experience a noticeable benefit from orthogonality at  $t = 9$  [s], at which time a  $90^\circ$  separation angle prevented the peak cost from exceeding  $30$  [m<sup>2</sup>]. One will also notice that though the UAVs do not hold a  $90^\circ$  separation angle  $\gamma$  continually, the separation angle does intermittently remain close to  $90^\circ$  for moderate lengths of time, e.g., when  $t \in [30, 40]$  and  $t \in [83, 95]$ . Accordingly, the optimal trajectories exhibit angle coordination as well for this particular speed and altitude



**Figure 3.6:** Cost function  $g(z) = \text{trace}(\mathcal{P})$ , planar distances  $\rho_1$  and  $\rho_2$ , and separation angle  $\gamma(t)$  for a target speed of  $v = 4.5$  [m/s] and zero initial condition. The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

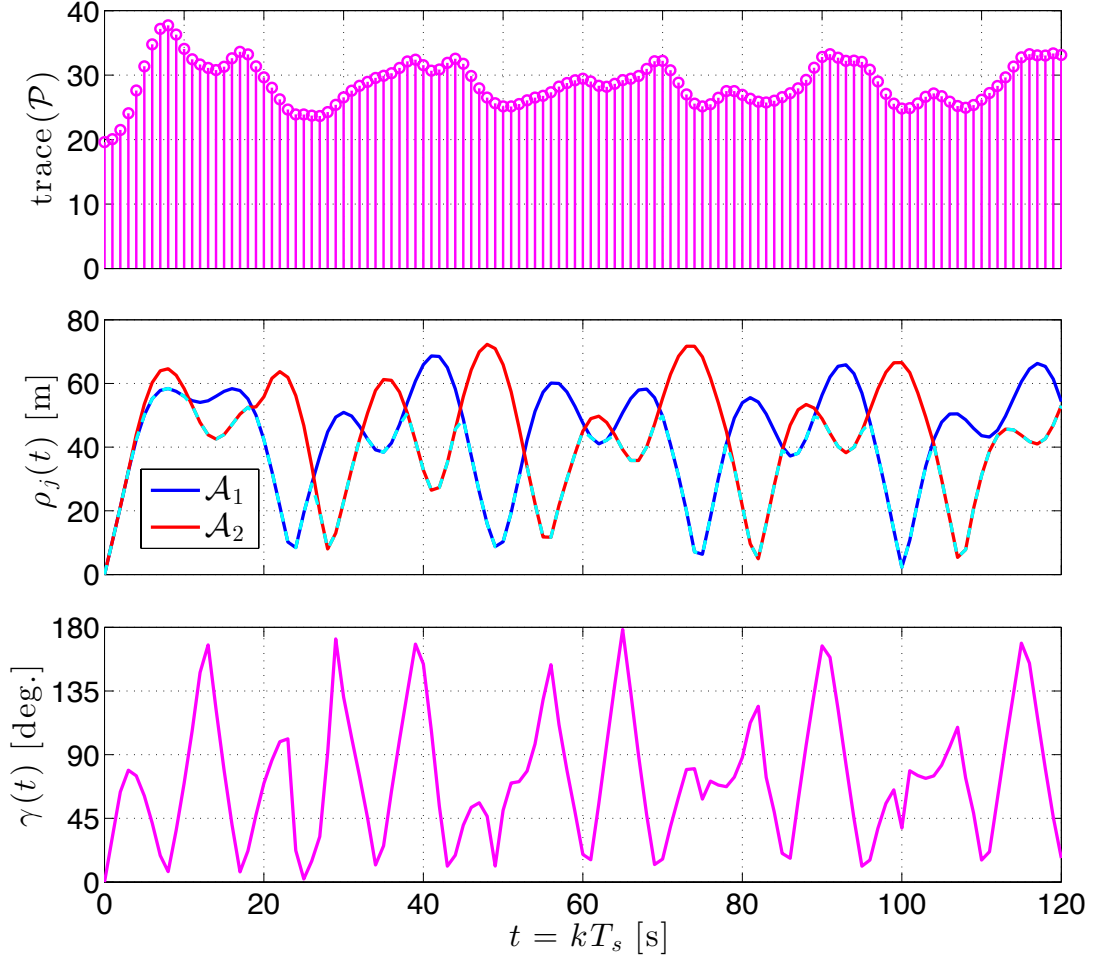
combination. Nonetheless, the cost function primarily resembles the minimum distance curve  $\varrho = \min\{\rho_1, \rho_2\}$ , though orthogonal viewing angles do offer some benefit for minimizing the fused GEC in this scenario.

To observe the effect of altitude on the optimally coordinated UAV trajectories, we again consider the slowest target speed of  $v = 4.5$  [m/s] but double



**Figure 3.7:** UAV trajectories over a two-minute window with a target speed of  $v = 4.5$  [m/s], an initial condition of  $z_0 = 0$ , and the baseline altitudes doubled. The UAV trajectories comprise quasi-periodic circular orbits of alternating direction that pass over or near to the target. The passes over the target are coordinated to be out of phase, as indicated by the distance curves of Figure 3.8.

each of the baseline UAV altitudes, i.e., we observe the optimal trajectories with  $h_1 = 80$  [m] and  $h_2 = 90$  [m]. The UAV trajectories are plotted in Figure 3.7 while the corresponding performance metrics are provided in Figure 3.8. By observing the trajectories, one should immediately notice how the UAVs no longer turn in the same direction, which is necessary to maintain a  $90^\circ$  separation angle. Rather, the UAVs make their loops in opposite directions, and hence we no longer observe the separation angle oscillating about  $\gamma = 90^\circ$ . However, the orbital trajectories remain the same in that they periodically pass over or near the target. Additionally, distance coordination yet again appears, as  $\mathcal{A}_2$  achieves its minimum relative distance at time instances close to those when  $\mathcal{A}_1$  achieves its peak relative distance, i.e., at times  $t \in \{49, 75, 100\}$ . The distance curves are



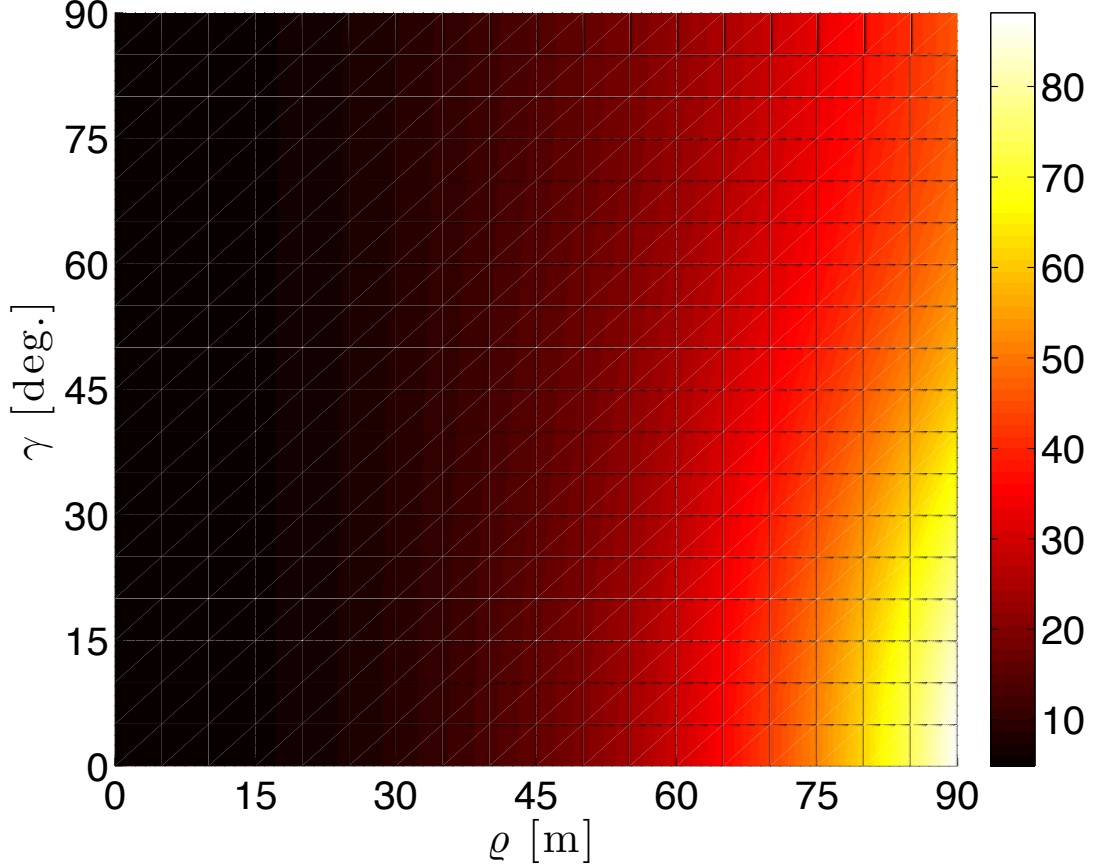
**Figure 3.8:** Cost function  $g(z) = \text{trace}(\mathcal{P})$ , planar distances  $\rho_1$  and  $\rho_2$ , and separation angle  $\gamma(t)$  with a target speed of  $v = 4.5$  [m/s], zero initial condition, and the baseline altitudes doubled. The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

quite out of phase for  $t = 23$  [s] as well. The final point we wish to make is that the fused GEC now has a higher bias since the individual GEC is a function of 3D distance. In fact, taking the average over the steady state costs, i.e., setting

$\bar{g} = (1/100) \sum_{k=21}^{120} g(z_k)$ , the lower altitude scenario yields a steady-state cost of  $\bar{g} \approx 13.12$  [m<sup>2</sup>] while the higher altitude scenario yields  $\bar{g} \approx 28.40$  [m<sup>2</sup>].

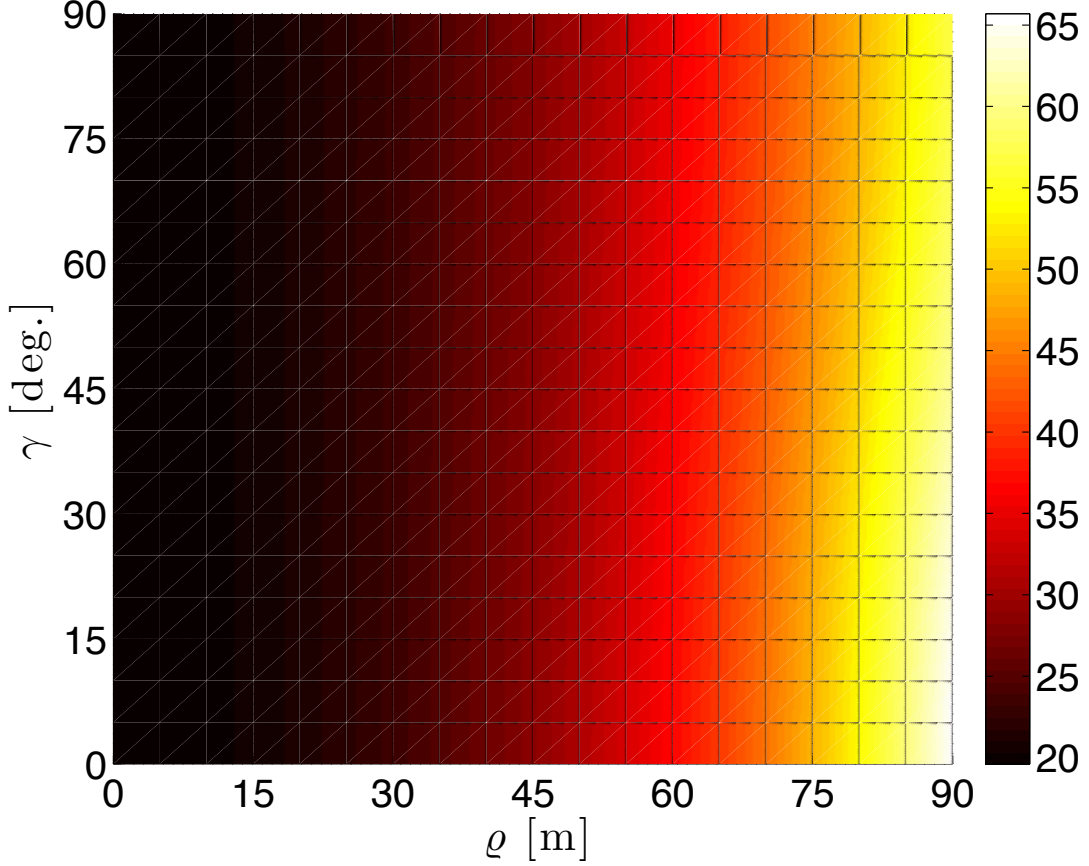
To understand why the importance of the separation angle vanishes with altitude, we have provided the plots of Figure 3.9 and 3.10 to be compared with one another, where the cost is plotted as a function of the separation angle  $\gamma$  and the minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  with  $\rho_1 = \rho_2$ . In Figure 3.9, the cost is computed with the baseline altitudes of Table 3.1 while in Figure 3.10 the cost is computed with the baseline altitudes doubled. From the surface plot of Figure 3.9, one can see how for  $\varrho \leq 45$  [m], the function decreases relatively little (less than 13%) as the separation angle increases from  $\gamma = 0^\circ$  to  $\gamma = 90^\circ$ . In the case of Figure 3.10, the function decreases less than 8% for  $\varrho \leq 75$  [m] as the separation angle increases from  $\gamma = 0^\circ$  to  $\gamma = 90^\circ$ . We note that at  $\varrho = 90$  [m], having orthogonal viewing angles ( $\gamma = 90^\circ$ ) offers a 48% reduction in the cost over collinear viewing angles ( $\gamma = 0^\circ$ ) at the baseline altitude; however, with the baseline altitudes doubled, orthogonality only offers a 13% reduction in the cost over collinear viewing angles for  $\varrho = 90$  [m]. Lastly, we point out that the minimum value of  $\text{trace}(\mathcal{P})$  is approximately 5 [m<sup>2</sup>] in Figure 3.9 and approximately 19.5 [m<sup>2</sup>] in Figure 3.10.

We now turn our attention to a scenario involving a slightly faster target speed of  $v = 5$  [m/s] and the original UAV altitudes of Table 3.1. UAV trajec-



**Figure 3.9:**  $\text{trace}(\mathcal{P})$  as a function of separation angle  $\gamma$  and minimum distance  $\rho := \min\{\rho_1, \rho_2\}$  with  $\rho_1 = \rho_2$  and with the altitudes of Table 3.1. Temporarily denoting the cost as  $\varphi(\gamma, \rho)$ , we note that  $\varphi(\gamma, \rho) = \varphi(\pi - \gamma, \rho)$ , where  $\gamma$  is in radians and  $\gamma \in [0, \pi/2]$ . Moreover, the cost function is reflective about  $\gamma = \pi/2$ , and hence we do not plot the cost for  $\gamma \in (\pi/2, \pi]$ .

tories are plotted in Figure 3.11 while corresponding performance metrics are provided in Figure 3.12. One will notice that, although the target speed has only increased by  $1/2$  [m/s] from the previous two scenarios, the UAVs no longer perform full circular orbits that periodically pass through the target. Rather, the agents perform out-of-phase sinusoidal trajectories, where the sinusoidal nature

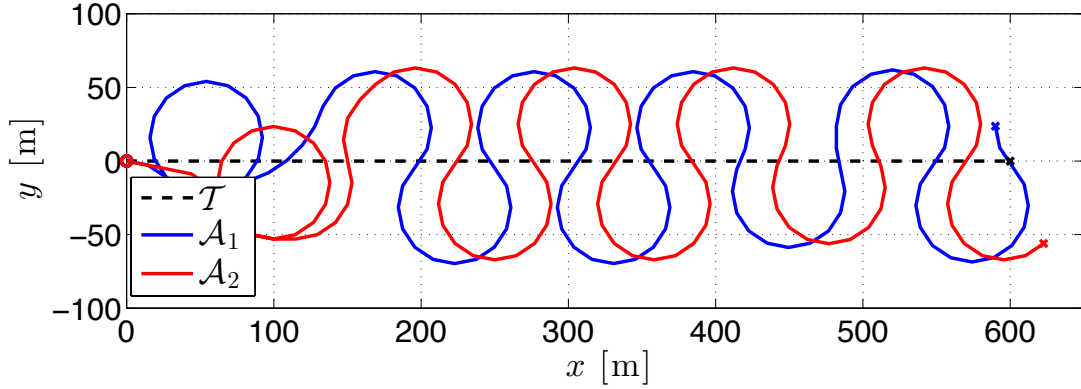


**Figure 3.10:**  $\text{trace}(\mathcal{P})$  as a function of separation angle  $\gamma$  and minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  with  $\rho_1 = \rho_2$  and with the baseline altitudes of Table 3.1 doubled. Temporarily denoting the cost as  $\varphi(\gamma, \varrho)$ , we note that  $\varphi(\gamma, \varrho) = \varphi(\pi - \gamma, \varrho)$ , where  $\gamma$  is in radians and  $\gamma \in [0, \pi/2]$ . Moreover, the cost function is reflective about  $\gamma = \pi/2$ , and hence we do not plot the cost for  $\gamma \in (\pi/2, \pi]$ .

is illustrated by Figure 3.11 while the phase is indicated by the distance curves of Figure 3.12. One will notice that, beginning at  $t = 30$  [s], the distance curves resemble full-wave rectified sinusoids that are out of phase. These particular trajectories enable the UAVs to approximately match their average speed along the direction of travel with that of the target, where the phase separation further

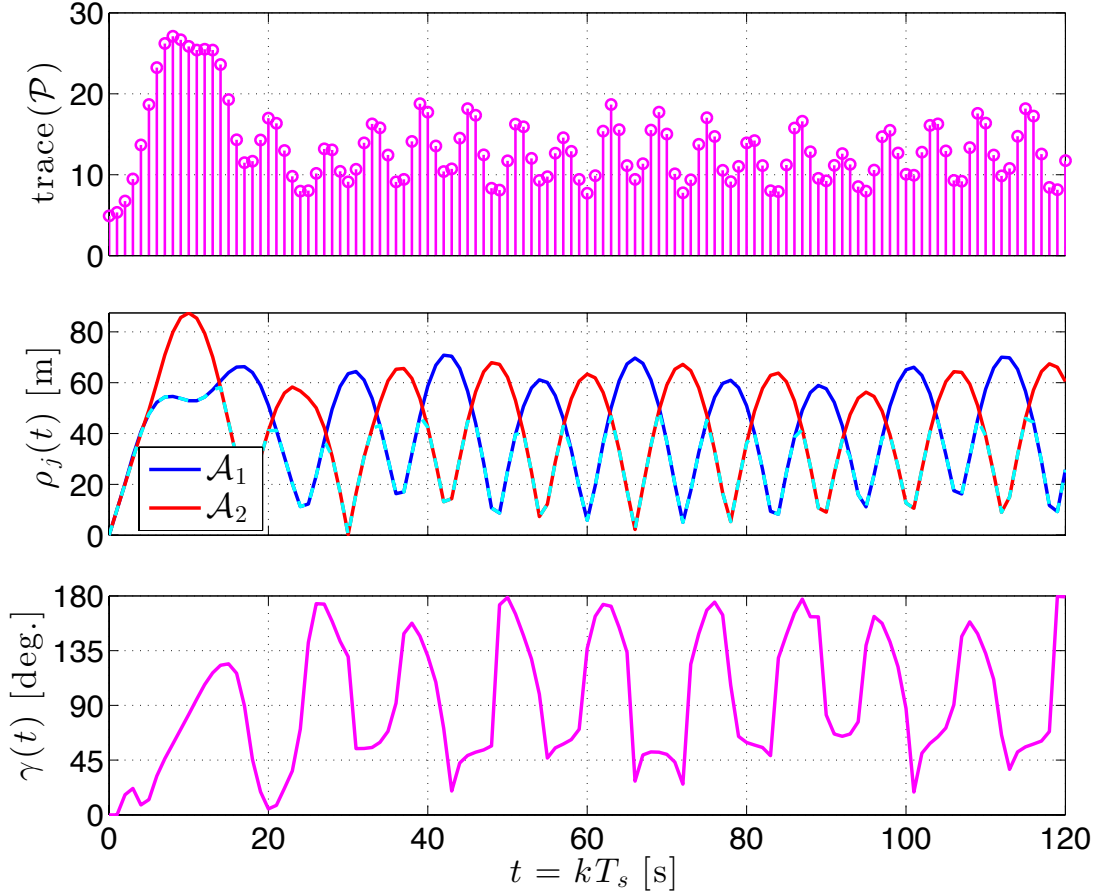


keeps the minimum distance from the target relatively small. In addition, the separation angle curve certainly passes through  $90^\circ$  a number of times, but does not hover near this value as in the case when  $v = 4.5$  [m/s] with the baseline altitudes. As the separation angle has less influence over the cost function at higher altitudes, we do not investigate a higher altitude scenario at this target speed.



**Figure 3.11:** UAV trajectories over a two-minute window for a target speed of  $v = 5$  [m/s] and an initial condition of  $z_0 = 0$ . The UAV trajectories comprise quasi-periodic sinusoidal orbits that pass over or near to the target. The passes over the target are coordinated to be out of phase, as indicated by the distance curves of Figure 3.12, as well as the terminal positions indicated by an “ $\times$ .”

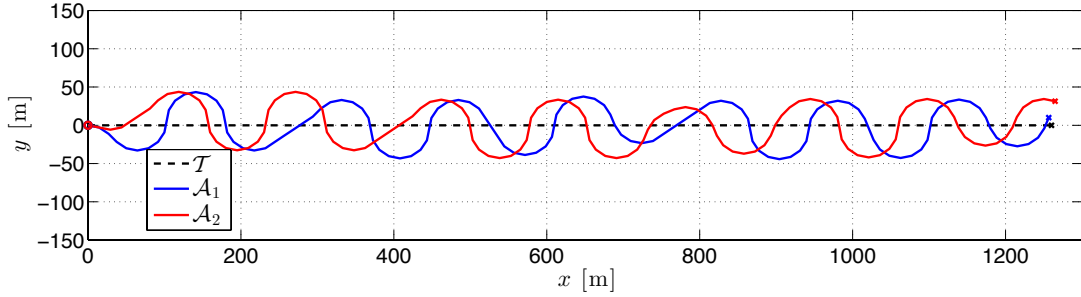
From the previous two scenarios wherein the altitudes were  $h_1 = 40$  [m] and  $h_2 = 45$  [m], the speed ratio  $\lambda = v/s$  at which the trajectories switch from circular orbits to sinusoids lies somewhere in between  $\lambda = 4.5/15 = 3/10$  and  $\lambda = 5/15 = 1/3$ . Coincidentally, in [3] Beard determines the optimal speed ratio ( $v/s$ ) at which a UAV (modeled as a Dubins vehicle) should switch between periodic orbital (or



**Figure 3.12:** Cost function  $g(z) = \text{trace}(\mathcal{P})$ , planar distances  $\rho_1$  and  $\rho_2$ , and separation angle  $\gamma(t)$  for a target speed of  $v = 5$  [m/s] and zero initial condition. The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

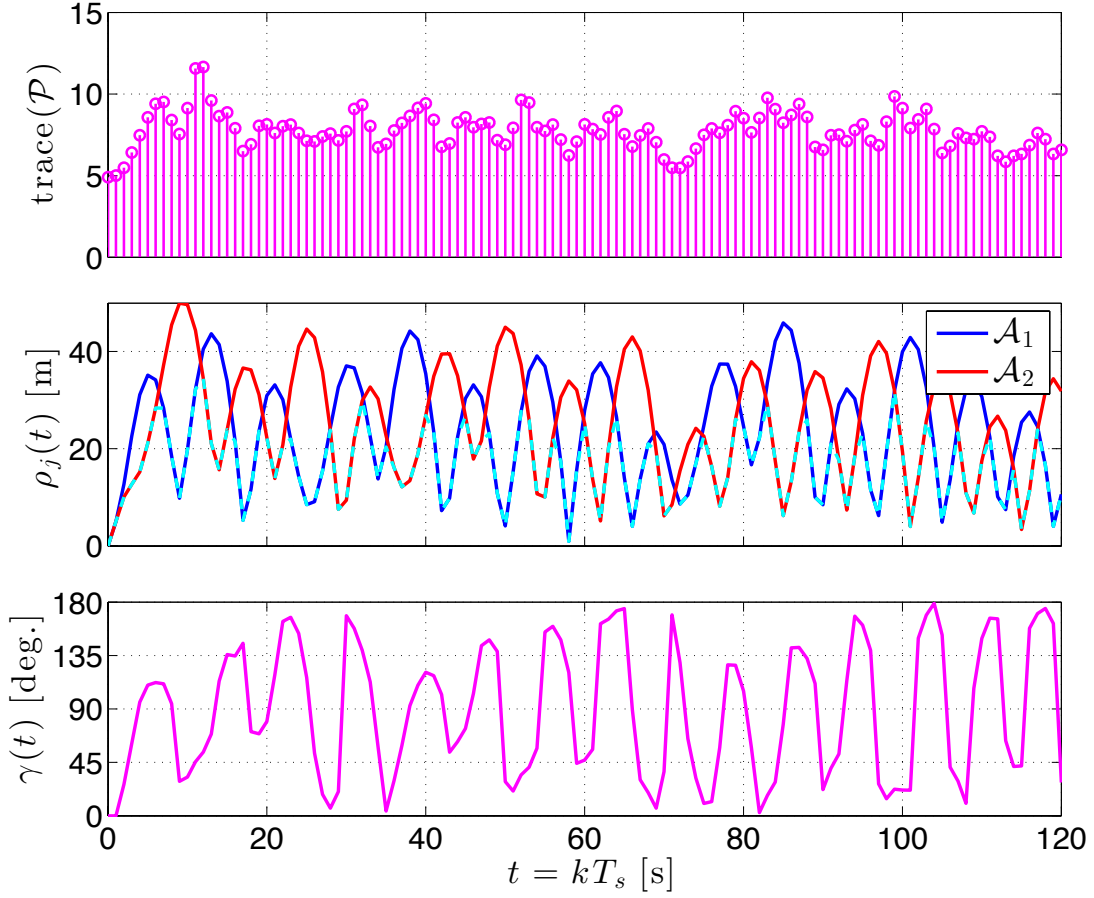
circular) trajectories and periodic longitudinal (or sinusoidal) trajectories so as to minimize the maximum deviation from the target along such periodic orbits. The optimal speed ratio for switching from orbital trajectories (preferred for lower target speeds) to sinusoidal trajectories (preferred for higher target speeds) was found to be  $\lambda^* = 0.3205$ . Of course, the orbital trajectories considered in [3]

do not intersect the target trajectory as they do here for  $v = 4.5$  [m/s] while Figure 3.6 shows that the maximum distance is not the most important feature of the optimal trajectories. Rather, the coordination of distances is the common key feature of the optimal behavior in the preceding two scenarios, as the cost function is primarily dependent on the minimum of the two distances relative to the target.



**Figure 3.13:** UAV trajectories over a two-minute window for a target speed of  $v = 10.5$  [m/s] and an initial condition of  $z_0 = 0$ . The UAV routes comprise sinusoidal trajectories that pass over or near to the target. The passes over the target are coordinated to be out of phase, as indicated by the distance curves of Figure 3.6, as well as the terminal positions indicated by an “ $\times$ .”

For the final scenario, we consider a speed ratio of  $\lambda = 10.5/15 = 7/10$  with the remaining parameters provided in Table 3.1. Moreover, we investigate the nature of the optimal trajectories and performance as the target speed approaches the UAV speed. UAV trajectories are plotted in Figure 3.13 while corresponding performance metrics are provided in Figure 3.14. The optimal UAV routes comprise sinusoidal trajectories with occasional straight-line paths



**Figure 3.14:** Cost function  $g(z) = \text{trace}(\mathcal{P})$ , planar distances  $\rho_1$  and  $\rho_2$ , and separation angle  $\gamma(t)$  for a target speed of  $v = 10.5$  [m/s] and zero initial condition. The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

that are most likely the result of the UAVs aiming to keep the peak distances out of phase, which can be seen in the plot of performance metrics. Thus, the routes again illustrate the UAVs aiming to match their average speed along the direction of travel with that of the target. In observing the performance metrics, one should notice how the cost is kept below  $10$  [m<sup>2</sup>] after  $t = 20$  seconds, which

cannot be said of the previous scenarios. The distance curves yet again have alternating peaks, which further corroborates the previous assertions of distance coordination. The separation angle  $\gamma$  certainly passes through  $\gamma = 90^\circ$  a number of times; however, the curve does not remain close to  $90^\circ$ . Again, since the separation angle has less influence over the cost function at higher altitudes, we do not investigate a higher altitude scenario at this target speed.

### 3.3.2 Comparison with Standoff Tracking

As noted in the related work, coordinated standoff tracking has been a long-standing goal in the general area of target tracking with a fair amount of work in recent years as well. As noted in [34], multiple fixed-speed UAVs cannot maintain a  $90^\circ$  separation angle at a fixed radius from a moving target. Hence, for the purpose of comparison, we focus on a splay-state controller that spreads UAVs uniformly in time along a periodic trajectory. The lower bound for the nominal standoff distance,  $\varrho_s$ , as imposed by a maximum roll angle  $\phi_{\max}$ , is given by Equation 5.37 in [21] as follows:

$$\varrho_s \geq \frac{(v + s)^2}{\alpha_g \tan \phi_{\max}}, \quad (3.35)$$

where  $v$ ,  $s$ , and  $\alpha_g$  denote target speed, UAV airspeed, and gravitational acceleration, respectively. To determine this upper bound in terms of a maximum turn

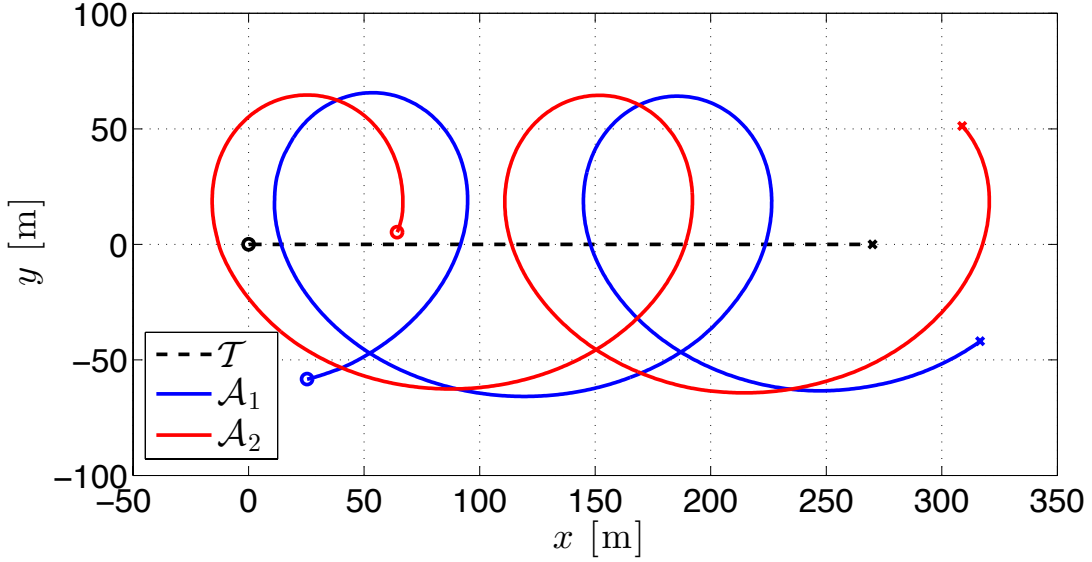
rate  $\bar{u}$ , one can use the relationship  $\bar{u} = (\alpha_g/v) \tan \phi_{\max}$  from (2.10) to obtain

$$\varrho_s \geq \frac{(v+s)^2}{s\bar{u}}. \quad (3.36)$$

With the target traveling at the slowest speed of 4.5 [m/s] and the remaining parameters given in Table 3.1, we have

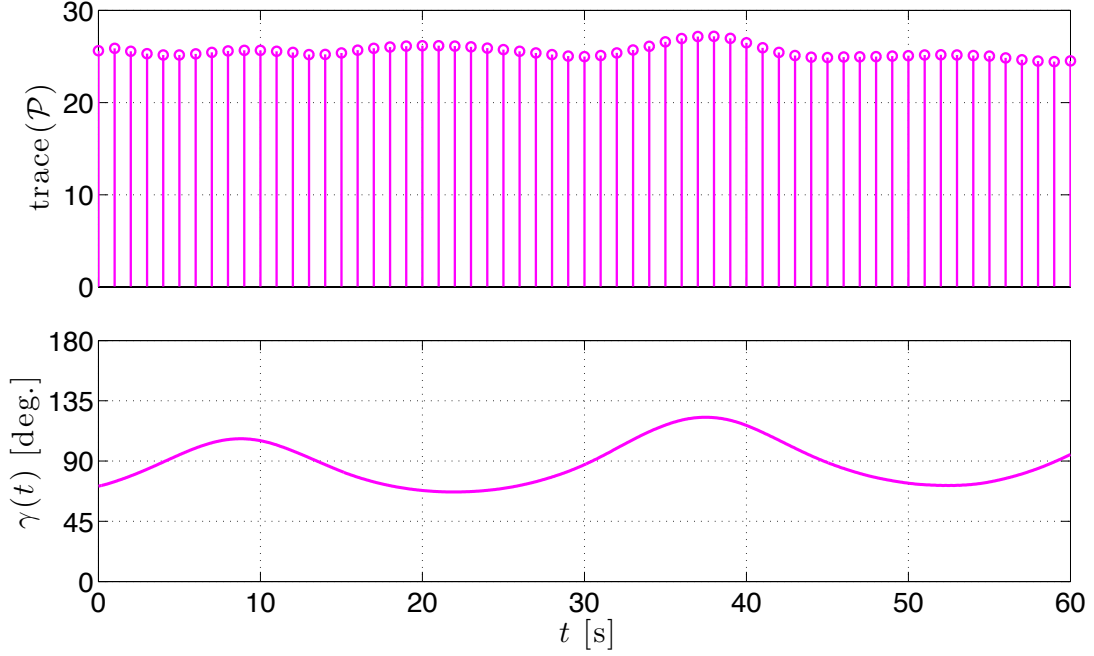
$$\varrho_s \geq \frac{(4.5 + 15)^2}{15 \cdot 2\pi/16} \approx 64.55 \text{ [m]}. \quad (3.37)$$

Hence we set  $\varrho_s = 65$  [m], and run simulations beginning in steady-state using the splay-state controller described in Chapter 5 of [21]. The trajectories and performance metrics are shown in Figures 3.15 and 3.16, respectively, with the sensor attitude covariance and UAV altitudes of Table 3.1. One can see that the separation angle oscillates around  $90^\circ$ , however, the cost exhibits only slight fluctuations about  $25.5 \text{ [m}^2\text{]}$ . In fact, if one evaluates the cost function at the ideal configuration of  $\rho_1 = \rho_2 = 65$  [m] and  $\gamma = 90^\circ$ , then  $\text{trace}(\mathcal{P}) \approx 25.5 \text{ [m}^2\text{]}$ . To obtain a fair comparison, we run the optimal control policy for  $v = 4.5$  [m] over a window of 5 minutes (to mitigate the effects of initial conditions) and set  $\bar{g} = (1/270) \sum_{k=31}^{300} g(z_k)$ . Moreover, we are truncating the effects of the initial conditions over the first 30 [s] in Figure 3.6 and averaging the cost over the remaining times to obtain the average steady-state cost  $\bar{g}$ , which is approximately  $13.3 \text{ [m}^2\text{]}$  in this case. Moreover, the optimal coordination strategy offers a near 48% reduction in the steady-state cost.



**Figure 3.15:** UAV trajectories with splay-state controller over a two-minute window for a target speed of  $v = 4.5$  [m/s] and a nominal distance of  $\varrho_s = 65$  [m]. The UAV trajectories comprise periodic circular orbits that are spread uniformly in time. The starting positions of all vehicles are indicated by an “o” while the ending positions are indicated by an “x.”

To gain additional insight into the significant gap between the two strategies, we revisit Figure 3.9 where the cost is plotted as a function of separation angle  $\gamma$  and minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  with  $\rho_1 = \rho_2$ . While the function changes relatively little with respect to the separation angle for  $\varrho \leq 45$  [m], at  $\varrho = 65$  [m], orthogonality does provides a significant advantage, as the cost is approximately  $25.5$  [m<sup>2</sup>] for  $\gamma = 90^\circ$  versus  $36$  [m<sup>2</sup>] for  $\gamma = 0^\circ$ . However, by observing Figures 3.6, 3.12, and 3.14, one can see how the steady-state ( $t \geq 20$  [s]) minimum distance varies roughly between 5 and 50 [m] with the optimal control policy. Thus, a  $90^\circ$  separation angle offers little advantage, even in the



**Figure 3.16:** Performance metrics for splay-state controller tracking a target target traveling at 4.5 [m/s], including cost  $g(z) = \text{trace}(\mathcal{P})$  at discrete times instances  $t \in \{0, 1, \dots, 60\}$  and separation angle  $\gamma(t)$  for  $t \in [0, 60]$ . The individual distances are not shown since they remained within three meters of the nominal standoff distance  $\varrho_s = 65$  [m].

higher range of the said distance interval. Moreover, while angle coordination is observed intermittently for  $v = 4.5$  [m/s] with the optimal coordination policy, this seems to have relatively little impact on the cost, as the cost rather seems to resemble the minimum distance curve. In any case, for faster target speeds, any semblance of angle coordination vanishes, as such a benefit is overshadowed by that of having the UAVs coordinate their distances to the target so as to keep the minimum distance to the target low.



One should notice that according to (3.36), the minimum standoff distance grows quadratically with the target speed  $v$ , and hence the cost function will increase significantly as well, since it is a nonlinear function of this minimum distance. Conversely, the average steady-state cost obtained with the optimal control policy decreases as the target speed increases (up to the UAV speed) as shown in Figure 3.14, as the UAV's are able to better match their average speed along the direction of travel with the target's speed by performing "S" turns or sinusoids over the target's path. Moreover, the performance gap will grow significantly for faster target speeds.

## 3.4 Conclusion

We have shown that for two *small* fixed-wing UAVs performing *vision-based* target tracking, the key optimal control strategy for gathering the best joint vision-based measurements is coordination of the distances to the target rather than pure viewing-angle coordination, which is the most prevalent control strategy for target tracking, even in recent years. This stems from the fact that fused geolocation error covariance is a function of the separation angle and the individual UAV distances from the target, where the minimum of the two distances is the dominant factor. Nonetheless, at the lowest target-UAV speed

ratio considered, the optimal control strategy comprised a mixture of both distance viewing angle coordination and distance coordination. However, by either increasing the altitude or increasing the speed ratio slightly, any semblance of angle coordination vanished.

In comparing the optimal controller with a standoff tracking controller at the lowest considered target speed, we found that the optimal controller outperformed the standoff tracking controller by nearly 48% in steady state. The reason for the large performance gap stems from the fact that the optimality criterion of fused geolocation error covariance is a function of three quantities, namely separation angle and individual planar distances from the target. A coordinated standoff tracking controller focuses on angle coordination at a fixed-nominal distance whose lower bound (set by the minimum turning radius) is proportional to the sum of the UAV and target speed squared. Moreover, for increasing target speed, the minimum allowable standoff distance increases quadratically. Since the minimum of the two UAV distances to the target is the dominant factor in the fused GEC, the cost of standoff tracking naturally increases nonlinearly with target speed. Thus, if one wishes to design scalable heuristic controllers for vision-based target tracking with small UAVs, one should focus on coordinating the distances to the target such that at least one UAV is never very far from the target.

## Chapter 4

# Stochastic Optimal UAV Coordination for Target Tracking

In the previous chapter, we saw that the optimal trajectories of two UAVs gathering the best joint vision-based measurements of a moving ground target were characterized by distance coordination, as the dominant factor in the fused geolocation error covariance is the minimum distance to target. Moreover, the UAVs coordinate their routes such that one UAV is never very far from the target. A number of heuristics have been proposed to approximate this optimal behavior at higher speeds using sinusoidal turn-rate control inputs. In particular, Lalish and Morgansen [22] propose oscillatory control of a constant-speed unicycle that generates sinusoidal trajectories whose average speed in the direction of travel matches that of a (piecewise) constant-velocity target. Furthermore, by setting a phase variable in the sinusoidal controller, one can have multiple unicycles generate sinusoidal trajectories that are out of phase, thereby emulating

distance coordination. In [38], Regina and Zanzi further refine the exposition of Lalish and Morgansen, provide an alternative outer-loop control algorithm that stabilizes the time-averaged system to the target trajectory, and corroborate the feasibility of the proposed control strategy by presenting simulation results using a 6-DoF flight simulator and a mild target motion. In the author's own investigation, the proposed oscillatory control strategies are non-robust to stochastic target motion. Additionally, roll dynamics are not considered while the use of a single type of trajectory to track the target can be quite suboptimal, as the previous chapter has shown that at slower target speeds circular trajectories (that occasionally pass over or near the target) are optimal while at higher speeds sinusoidal trajectories are optimal.

Of course, even the optimal control policies presented in the previous chapter were obtained for individual target speeds under ideal settings. In particular, the UAVs were modeled as Dubins vehicles with bang-off-bang turn-rate control while the target was modeled as a unicycle with a constant velocity. In general, UAVs are not only not capable of instantaneous changes in turn rate, but they are also subject to roll dynamics. Furthermore, even if one utilizes a fourth-order model as in Chapter 2, a real UAV will exhibit phenomenon not captured by the model due to the reduction from the fully realistic 6-DoF model. Additionally, environmental disturbances, e.g., wind gusts, will almost certainly

be encountered as well. Another important factor from a pragmatic standpoint is the fact that real target motion is unpredictable. By formulating the optimal coordination problem in a stochastic framework as in Chapter 2, one can obtain a control policy that optimally coordinates two UAVs with roll dynamics while being robust to stochastic target motion, environmental disturbances, and unmodeled UAV dynamics.

The purpose of this chapter is thus to present and study an effective solution to the problem of optimally coordinating two UAVs to track a moving ground target under fairly realistic conditions, thereby remedying all of the limiting assumptions listed at the end of Section 3.1.1. Namely, two camera-equipped UAVs must gather the best *joint* vision-based measurements of a randomly moving ground target whilst themselves being subject to limited control effort and experiencing stochasticity in their dynamics. Due to the success of the stochastic approach in Chapter 2, we will utilize fourth-order stochastic models similar to those in that chapter and formulate the problem in the framework of stochastic optimal control. This will produce a control policy robust to real-world phenomenon and will allow us determine what aspect of the results will remain the same under realistic settings. Of course, utilizing fourth-order models for all vehicles renders the grid-based basic Monte Carlo solution of Chapter 2 intractable, and hence a significant portion of this chapter will be devoted to presenting a

regression-based dynamic programming solution to the problem. As the literature review from the previous chapter on coordinated target tracking pertains to the present discussion, we immediately present the problem formulation.

## **4.1 Problem Formulation**

Consider a group of two UAVs tasked with autonomously tracking an unpredictable moving target vehicle using gimbaled video sensors. The UAVs fly at a constant altitude and fixed nominal airspeed yet experience stochasticity in their dynamics. The target is a nonholonomic ground vehicle that moves on the ground and exhibits stochastic turning and acceleration. The main objective is to optimize the coordination of the UAVs to gather the best joint vision-based measurements of the target. Since all vehicles experience stochasticity in their dynamics, the dynamic optimization is inherently a stochastic optimal control problem, whose key components are a description of the stochastic evolution of the states and the cost associated with each state. As the cost function is the geolocation error covariance described in Section 3.2.2, we need only describe the stochastic kinematic models for both the UAVs and target along with the corresponding state space description.

### 4.1.1 Stochastic Vehicle Dynamics

We utilize the UAV and target state definitions of Section 2.2.1 and assume they evolve stochastically according to discrete-time Markov Decision Processes, as in the stochastic optimal control design of Section 2.3. Thus, for  $j \in \{1, 2\}$ , the probability of transitioning from the current UAV state  $\xi_j$  to the next UAV state  $\xi'_j$  under the change  $u_j$  in  $r_j$  is given by the controlled state transition probability function  $p_a(\xi'_j | \xi_j, u_j)$ . Moreover, we utilize the same state transition probability for both UAVs, though one can certainly consider heterogeneous teams. Furthermore, throughout this chapter, we shall denote by  $j \in \{1, 2\}$  the UAV index, and hence all UAV-related quantities from Chapter 2 are subscripted with  $j$  to distinguish the individual agents. The probability of transitioning from the current target state  $\eta$  to the next target state  $\eta'$  is given by the state transition probability function  $p_g(\eta' | \eta)$ .

As in Section 2.3, we develop empirical characterizations of the individual agent kinematics that allow one to draw Monte Carlo samples  $\tilde{\xi}_1^{(i, u_1)}$ ,  $\tilde{\xi}_2^{(i, u_2)}$ , and  $\tilde{\eta}^{(i)}$ , where  $i \in \{1, 2, \dots, N_p\}$ , from appropriate conditional probability density functions rather than derive explicit formulas for state transition probabilities. To generate sample trajectories of the combined state  $z$ , one simply combines  $\tilde{\xi}_1^{(i, u_1)}$ ,  $\tilde{\xi}_2^{(i, u_2)}$ , and  $\tilde{\eta}^{(i)}$  according to transformations (similar to equations (2.15) and (2.16)) that place the UAVs in a 9-dimensional target-centric state space  $\mathcal{Z}$ .

Thus, with  $u = (u_1, u_2)$  and  $z \in \mathcal{Z}$ , this provides an empirical characterization of the overall stochastic dynamics that evolve according to a controlled state transition probability function  $p(z' | z, u)$ . The ability to sample this state transition probability will suffice to effectively approximate the dynamic programming solution.

We utilize the stochastic kinematic model for the UAV given in Section 2.3.2 with the exception that the stochastic airspeed is now drawn from a symmetric triangle distribution rather than a normal distribution. This distribution is also characterized by its mean  $\bar{s}$  and variance  $\sigma_s^2$ . As we are now working with 2 UAVs, we take  $r := (r_1, r_2)$  and  $u := (u_1, u_2)$ . Since, for  $j \in \{1, 2\}$ ,  $r_j \in C$ , where  $C$  is given by (2.3), we have  $r \in C^2$ . Also, since  $u_j \in U(r_j)$ , with  $U(r_j)$  given by (2.7), we have  $u \in \mathcal{U}(r)$ , where this new 2-dimensional roll-angle action space is given by

$$\mathcal{U}(r) := U(r_1) \times U(r_2). \quad (4.1)$$

The initial continuous-time kinematic model for the target is that of a planar kinematic unicycle with turn rate and acceleration control inputs and is given by (2.11). We again model the behavior of an operator driving the ground vehicle safely and casually, yet unpredictably; however, we now use a more refined model that generates milder and arguably more realistic target motion. This



model also has the advantage that the designer can set the bounds on the target speed such that the target speed both naturally and stochastically evolves within these bounds. These bounds are chosen to keep the problem both interesting (no stopped targets) and well posed (the maximum target speed less than the UAV's airspeed). As in Section 2.3.3, the control inputs for the target are drawn from continuous probability density functions, and they are assumed to be held constant over a  $T_s$ -second ZOH period synchronized with the UAV's.

Concerning the target's speed, we assume that it is bounded below by  $\underline{v}$  and above by  $\bar{v}$ , where the former quantity is nonnegative and the latter is strictly less than the UAV's airspeed  $s_a$ . Since the target's speed at the end of the ZOH period is  $v' = v + aT_s$ , we enforce the speed bounds, as well as a maximum acceleration  $\alpha > 0$ , by assuming the acceleration is drawn from a symmetric triangle distribution with lower bound  $\underline{a} = \max\{(\underline{v} - v)/T_s, -\alpha\}$  and upper bound  $\bar{a} = \min\{(\bar{v} - v)/T_s, \alpha\}$ .

We assume the target vehicle is subject to a minimum turning radius  $\varrho_g > 0$ , and we take the corresponding maximum turn rate to be  $\bar{\omega}_\varrho = \min\{v/\varrho_g, (v + aT_s)/\varrho_g\}$ . Furthermore, if the maximum allowable turn rate is  $\bar{\omega}_a$ , then we take the overall maximum turn rate to be  $\bar{\omega} = \min\{\bar{\omega}_\varrho, \bar{\omega}_a\}$ . The quantity  $\bar{\omega}_a$  is typically less than  $\bar{\omega}_\varrho$  at moderate to high speeds and is used to further govern the target's turning behavior beyond the inherent minimum turning-radius lim-

**Table 4.1:** Stochastic target motion parameters

Parameter:	$\alpha$	$\underline{v}$	$\bar{v}$	$\underline{\varrho}_g$	$\bar{\omega}_a$	$T_s$
Value:	0.5	4.5	12.5	7	0.2	2
Units:	m/s <sup>2</sup>	m/s	m/s	m	rad./s	s

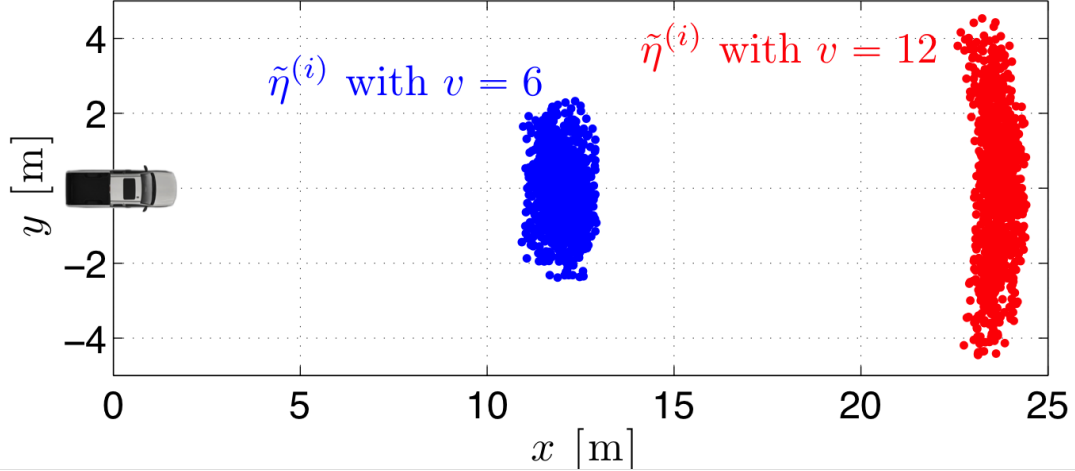
itation. We then take the target's turn rate to be distributed according to a symmetric triangle distribution with lower bound  $-\bar{\omega}$  and upper bound  $\bar{\omega}$ . Finally, the discrete-time stochastic kinematic model is the solution of (2.11) at the end of the  $T_s$ -second ZOH period with the acceleration and turn rate having been drawn from their respective triangle distributions at the start of the ZOH period. This kinematic model is illustrated in Figure 4.1 with the parameters in Table 4.1.

### 4.1.2 Target-Centric State Space

We consider a 9-dimensional target-centric state space  $\mathcal{Z}$ . For  $j \in \{1, 2\}$ , we denote by  $\mathbf{r}_j$  the relative position of UAV  $j$ , which is given by

$$\mathbf{r}_j := \begin{bmatrix} \cos \psi_g & \sin \psi_g \\ -\sin \psi_g & \cos \psi_g \end{bmatrix} \begin{bmatrix} x_{a,j} - x_g \\ y_{a,j} - y_g \end{bmatrix}. \quad (4.2)$$

Also, as in the previous chapter, we define the UAV  $j$ 's pose (position and heading) relative to the target as  $\mathbf{p}_j := (\mathbf{r}_j, \psi_{r,j}) \in \mathbb{R} \times [-\pi, \pi)$ , where  $\psi_{r,j} = \text{atan2}(\sin(\psi_{a,j} - \psi_g), \cos(\psi_{a,j} - \psi_g))$  and  $\text{atan2}$  is the four-quadrant inverse tangent



**Figure 4.1:** Sample positions generated from the stochastic target motion model. The two initial target states depicted with different colors correspond to identical initial positions at the origin, but two distinct initial speeds of 6 and 12 [m/s]. For each initial condition, 1,000 samples are generated.

function. The state vector  $z \in \mathcal{Z} \subset \mathbb{R}^9$  is thus given by

$$z := (\mathbf{p}_1, r_1, \mathbf{p}_2, r_2, v). \quad (4.3)$$

The overall state transition probability  $p(z' | z, u)$ , where  $u \in \mathcal{U}(r)$ , is given implicitly by combining the stochastic kinematic models for the vehicles with the preceding description of the components of the states in  $\mathcal{Z}$ .

### 4.1.3 Stochastic Optimal Control Objective

The stochastic optimal coordination problem is to determine the optimal feedback control policy  $\mu_k^* : \mathcal{Z} \rightarrow \mathcal{U}^2$ ,  $k \in \{0, 1, \dots, K-1\}$ , that minimizes

$$J(z) = \mathbb{E} \left[ \sum_{k=0}^K g(z_k) \middle| z_0 = z \right], \quad \forall z \in \mathcal{Z}, \quad (4.4)$$

where  $z_k = z(kT_s)$ ,  $K \in \mathbb{N}$ ,  $\mathbb{E}[\cdot]$  denotes expectation,  $g(\cdot)$  is given by (3.28),  $\mathcal{U}$  is given by (2.8), and  $z_{0:K}$  is a Markov Decision Process that evolves according to the transition probability  $p(z' | z, u)$  determined by the models described in Section 4.1.1 and the state definitions in Section 4.1.2, under the feedback law  $u_k = \mu_k^*(z_k)$ . Note that the state transition probability  $p(z' | z, u)$  can also be written as  $p(z_{k+1} | z_k, u_k)$ . Henceforth throughout this chapter, for some  $w \in \mathbb{R}^n$ , we shall denote by  $w_{0:K}$  the ordered sequence of vectors  $(w_0, w_1, \dots, w_K)$ . To solve this problem, we present a novel optimal policy generation algorithm based upon the policy iteration technique known as regression Monte Carlo.

The basic Monte Carlo method of Section 2.3.4 method is suitable for smaller stochastic optimal control problems, such as the single-UAV target tracking scenario for which the state dimension  $n$  is 5, as demonstrated in Chapter 2. We will however utilize this method later to generate a baseline control strategy for comparison in which two UAVs are performing target tracking independently. However, such a grid-based solution is simply infeasible for larger state spaces,

such as that corresponding to the present two-UAV scenario wherein the state dimension is  $n = 9$ ; as a result, one must employ more sophisticated techniques to determine an approximately optimal policy. Our goal is not to provide a broad overview of possible solution techniques, but rather to focus on a particular method that is effective for solving the present discrete-time, discrete-decision, stochastic optimal control problem over a finite planning horizon. Thus we simply note that [4] and [37] are textbooks addressing multiple classes of stochastic optimal control problems and proceed to describe the simulation-based policy iteration technique known as regression Monte Carlo, which is well-suited for the present application and further adapted into a *policy generation* algorithm to remove the need for an initial policy map.

## 4.2 Regression Monte Carlo

Regression Monte Carlo (RMC) is a simulation-based policy iteration algorithm that was introduced to stochastic control by Longstaff and Schwartz [27] in the context of optimal stopping and more formally by Egloff [12]. It is suitable for moderate dimensional stochastic optimal control problems, e.g., those having state dimension  $1 - 10$ , wherein one may not have an analytic expression for the state transition probability but can easily generate samples. The power and

versatility of RMC is underscored by its use in determining optimal policies for managing influenza outbreaks in [28], as well as optimal policies for autonomous vehicle coordination in the current setting. Here we present the method in the general setting following the description of [28].

### 4.2.1 Standard Technique

This work utilizes the  $Q$ -value (referred to as the *continuation cost* in [28], or perhaps more commonly as the  $Q$ -factor [4]), which is defined as

$$Q_k(z, u) := \min_{u_{k+1:K-1}} \mathbb{E} \left[ \sum_{\ell=k}^K g(z_\ell) \middle| z_k = z, u_k = u \right]. \quad (4.5)$$

The  $Q$ -value is the expected cumulative (or pathwise) cost of being at a state  $z$  at time  $k$ , applying control action  $u \in U(z)$  at that time, and then applying an optimal policy from time  $k+1$  onward. Here,  $U(z)$  denotes the state-dependent action space, which we assume is finite for all  $z \in \mathcal{Z}$ . Also note that since for  $t \in \{k+1, k+2, \dots, K-1\}$ ,  $u_t$  is a feedback policy, i.e.,  $u_t = \mu_t(z_t)$ , the optimization is not over a fixed sequence but over the sequence of mappings  $\{\mu_t(z_t)\}_{t=k+1}^{K-1}$ .

Accordingly, the  $Q$ -value and the value function of Section 2.3.4 are related as follows:

$$\begin{aligned} Q_k(z, u) &= g(z) + \mathbb{E}[V_{k+1}(z') | z, u] \\ &= g(z) + \int V_{k+1}(z') p(z' | z, u) dz', \end{aligned} \quad (4.6)$$

and

$$V_k(z) = \min_{u \in U(z)} Q_k(z, u). \quad (4.7)$$

Thus, the optimal control policy is also formed as

$$\mu_k^*(z) = \arg \min_{u \in U(z)} Q_k(z, u). \quad (4.8)$$

The main idea of RMC methods is to determine  $\mu_k^*(z)$  from (4.8) for  $k \in \{K-1, K-2, \dots, 0\}$ , by approximating  $Q_k(z, u)$  through Monte Carlo simulations of the right-hand-side of (4.6) for each  $z \in \mathcal{Z}$  and for each  $u \in U(z)$ . This is done on a stochastic mesh corresponding to a collection of simulated paths  $\{z_{0:K}^{(i)}\}$ , where  $i \in \{1, 2, \dots, M\}$ , that are generated with initial policy  $\mu_k^{(0)}(z)$  starting from a collection of initial conditions  $\{z_0^{(i)}\}$ . In certain RMC applications, one is interested in starting from a single initial condition, in which case the initial condition set contains  $M$  copies of the same initial state. However, in a receding-horizon application wherein one always implements  $\mu_0^*(z)$ , one will need to estimate  $Q_0(z, u)$  over a considerable portion of the state space. Accordingly,

$\{z_0^{(i)}\}$  may comprise samples of a distribution over the state space that reflects where steady state trajectories are likely to evolve. In any case, each trajectory  $z_{0:K}^{(i)}$  is a realization of the Markov Decision Process  $z_{0:K}$  with initial condition  $z_0^{(i)}$  and feedback law  $\mu_k^{(0)}(z)$ . Also, to simplify the introductory discussion of RMC, we will assume for now that the control action space  $U(z)$  is the same for all  $z \in \mathcal{Z}$  and hence refer to it simply as  $U$ .

In a basic Monte Carlo method, one estimates the expectation in (4.6) by generating  $N_s$  scenarios for a given state  $z$  and control action  $u$  at time  $k$  and then taking the empirical average. For example, with  $V_K(z) = g(z)$ , one can estimate  $Q_{K-1}(z, u)$  as follows

$$Q_{K-1}(z, u) \approx g(z) + \frac{1}{N_s} \sum_{i=1}^{N_s} g(\tilde{z}^{(i,u)}), \quad (4.9)$$

where the sample point is taken from the state transition probability function, i.e.,  $\tilde{z}^{(i,u)} \sim p(z' | z, u)$ . While this is reasonable for a single point, it would be impractical to do so for each control action in  $U$  and point  $z_{K-1}^{(i)}$  in the stochastic mesh, as this would require  $N_u M N_s$  Monte Carlo simulations, where  $N_u = |U|$ . Thus, in RMC, one simply generates a single realization of the cumulative cost for each point in the stochastic mesh and for each control action in  $U$  and then carries out cross-sectional regression to fit the *entire* map  $(z, u) \mapsto Q_{K-1}(z, u)$ . Moreover, for each  $u \in U$ , one generates a cumulative cost realization  $\tilde{q}_{K-1}^{(i,u)}$  given



by

$$\tilde{q}_{K-1}^{(i,u)} = g\left(z_{K-1}^{(i)}\right) + g\left(\tilde{z}_K^{(i,u)}\right), \quad (4.10)$$

and then regresses  $\{\tilde{q}_{K-1}^{(i,u)}\}$  onto statistics derived from  $\{z_{K-1}^{(i)}\}$  in order to generate an approximation to the corresponding continuation cost  $\hat{Q}_{K-1}(z, u)$ . As the regression step is crucial to the performance of RMC, this will be addressed in one of the following sections.

The new policy at time  $K - 1$  is then given by

$$\mu_{K-1}^{(1)}(z) = \arg \min_{u \in U} \hat{Q}_{K-1}(z, u) \quad (4.11)$$

Also, while one might be inclined to take

$$\hat{V}_{K-1}(z) = \min_{u \in U} \hat{Q}_{K-1}(z, u) \quad (4.12)$$

and repeat the same procedure for  $k = K - 2$  by substituting  $\hat{V}_{K-1}(z)$  for  $V_{K-1}(z)$  in (4.6), this practice generally leads to rapid error accumulation. Rather RMC focuses on approximating the policy map  $\mu_k^{(1)}(z)$  rather than the continuation cost. More specifically, at time  $k$  one re-simulates a single trajectory  $\tilde{z}_{k+1:K}^{(i,u)}$  for each point  $z_k^{(i)}$  in the stochastic mesh using control action  $u$  at time  $k$  and implementing future controls based on the newly constructed policy map  $\mu_t^{(1)}(z)$ , where  $t \in \{k + 1, k + 2, \dots, K - 1\}$ . Summing the associated stage costs yields  $\tilde{q}_k^{(i,u)}$ , which is a realization of the pathwise cost and is exact for this scenario with

the exception of incorrect future policy decisions. Therefore, the approximation of (4.6) only contributes an error when it provides an incorrect ranking of future optimal control actions.

The general algorithm for  $k \in \{0, 1, \dots, K\}$  is given by Algorithms 4.1 and 4.2 and has an overall computational complexity of  $\mathcal{O}(K^2 M N_u)$ . In standard RMC the set of initial trajectories  $\{z_{0:K-1}^{(i)}\}$  required by Algorithm 4.1 is generated by applying the initial control policy  $\mu_k^{(0)}(z)$  to some set of initial conditions  $\{z_0^{(i)}\}$ . Note that in the algorithms we refer to the realizations of state vectors at a particular time as *particles*. Also, in a typical implementation, the inner **for** loop of Algorithm 4.1 is computed in parallel while the outermost **for** loop of Algorithm 4.2 is a vectorized operation. Two key components of the algorithm must be selected to obtain acceptable performance, namely the initial policy map and the regression type used.

### 4.2.2 Modified Technique for Policy Generation

As with any regression, a higher concentration of scenarios in a given neighborhood improves the prediction accuracy therein. Hence, one major source of influence on the performance of the resulting policy map is the initial policy map  $\mu_k(z)$ , since it steers the stochastic evolution of the states to generate the initial trajectory set  $\{z_{0:K}^{(i)}\}$ . Therefore, an initial policy map close to the optimal will

---

**Algorithm 4.1** Standard Regression Monte Carlo

---

**Require:** Set of initial trajectories  $\{z_k^{(i)}\}$  for  $i \in \{1, 2, \dots, M\}$  and  $k \in \{0, 1, \dots, K-1\}$

- 1:  $N_u \leftarrow |U|$
- 2: **for**  $k = K-1, K-2, \dots, 0$  **do**
- 3:      $Z \leftarrow \{z_k\}$
- 4:     **for**  $\ell = 1, 2, \dots, N_u$  **do**
- 5:         Using Algorithm 4.2, generate cumulative cost realization vector  $q \in \mathbb{R}^M$  corresponding to control action  $u^{(\ell)} \in U$
- 6:         Regress  $q_i$ 's against statistics derived from each  $\zeta^{(i)} \in Z$  to determine  $\hat{Q}_k(z, u^{(\ell)})$
- 7:     **end for**
- 8: **end for**
- 9: **return**  $Q$ -value approximators  $\hat{Q}_k(z, u)$ , where  $k \in \{0, 1, \dots, K-1\}$

---



---

**Algorithm 4.2** Generate a single realization of the pathwise cost for each particle in  $Z$ , where control action  $u^{(\ell)}$  is applied to each particle at time  $k$

---

**Require:** Particle set  $Z \subset \mathcal{Z}$ ; control action  $u^{(\ell)} \in U$ ; time index  $k$ ;  $\hat{Q}_{k+1:K-1}(z, u)$  if  $k \leq K-2$ ;

- 1:  $M \leftarrow |Z|$
- 2: **for**  $i = 1, 2, \dots, M$  **do**
- 3:     Sample  $\tilde{\zeta}^{(i)} \sim p(z' | \zeta^{(i)}, u^{(\ell)})$ , where  $\zeta^{(i)} \in Z$
- 4:      $q_i \leftarrow g(\zeta^{(i)}) + g(\tilde{\zeta}^{(i)})$
- 5:     **if**  $k+1 < K$  **then**
- 6:         **for**  $t = k+1, k+2, \dots, K-1$  **do**
- 7:              $\zeta^{(i)} \leftarrow \tilde{\zeta}^{(i)}$
- 8:              $u^* = \arg \min_{u \in U} \hat{Q}_t(\zeta^{(i)}, u)$
- 9:             Sample  $\tilde{\zeta}^{(i)} \sim p(z' | \zeta^{(i)}, u^*)$
- 10:             $q_i \leftarrow q_i + g(\tilde{\zeta}^{(i)})$
- 11:         **end for**
- 12:     **end if**
- 13: **end for**
- 14: **return**  $q \in \mathbb{R}^M$

---

lead to re-simulation trajectories  $\{\tilde{z}_{k+1:K}^{(i,u)}\}$  in Algorithm 4.2 that lie close to the original trajectory set where the prediction accuracy is highest; otherwise, one will be forced to perform *extrapolation* with the  $Q$ -value approximators  $\hat{Q}_k(z, u)$ , which may lead to large errors. Thus, to circumvent the need and influence of an initial policy map, we propose a policy *generation* technique.

The main idea for the new technique is to assume that for some set  $Z \subset \mathcal{Z}$  consisting of  $M$  particles, the majority of trajectories corresponding to the optimal controller  $\mu^*(z)$  will always remain close to this set in some sense. Thus, we take the initial set of trajectories in Algorithm 4.1 to be equal to this set at each time step, i.e.,  $\forall k \in \{0, 1, \dots, K-1\}, \{z_k^{(i)}\} = Z$ . This avoids extrapolation in the regression-based prediction of the continuation cost, and hence, in principle, the prediction accuracy should remain sufficient for choosing the correct control action. Moreover, with intuition and insight into the problem, one can construct  $Z$  to have a majority of the particles near the steady-state optimal trajectories. In this work, we refer to  $Z$  as the *initial condition* set, which refers to that fact that all re-simulations in Line 5 of Algorithm 4.1 start from the same locations and not that the simulations are beginning from  $k = 0$ . Nonetheless, with regard to Line 3 of Algorithm 4.1,  $\{z_k\} = \{z_0\}, \forall k \in \{1, 2, \dots, K-1\}$ . We take the initial condition set to be randomly generated at the start of RMC according to some distribution over the state space. One may also generate deterministic grids, as

in Section 2.3.4; however, the dimensionality of the problem may hinder such an approach.

### 4.2.3 Regression

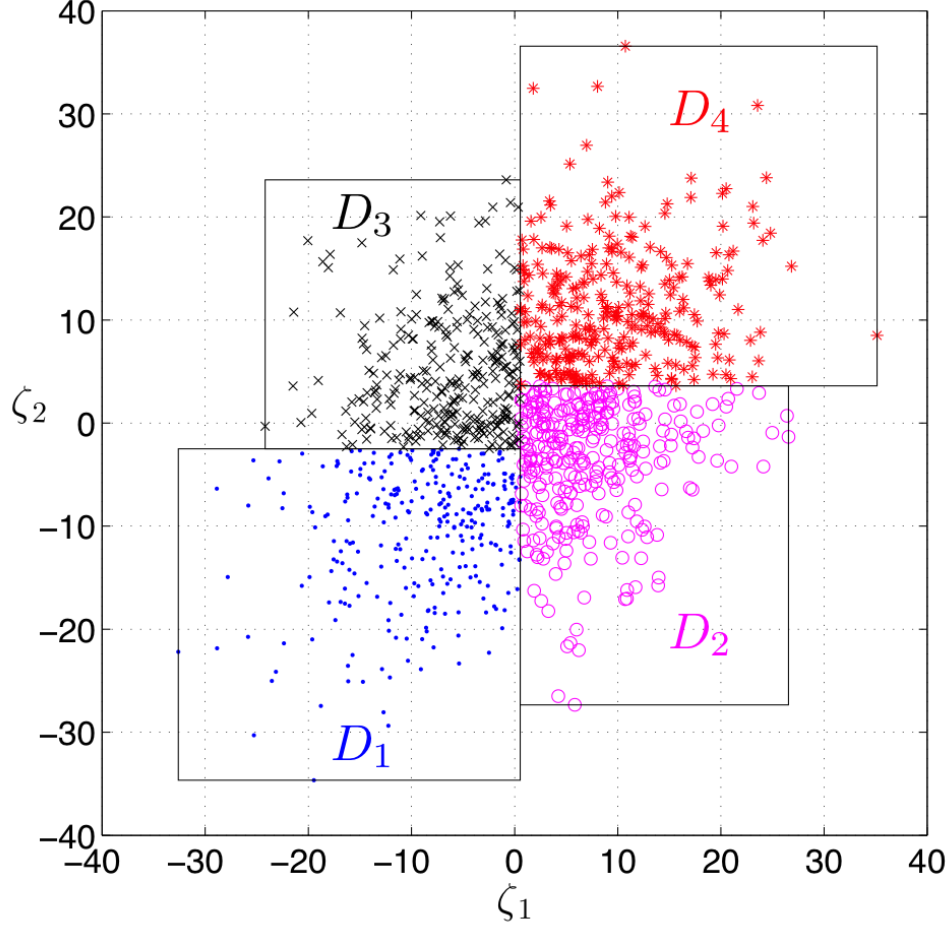
The regression type used is also crucial to the performance of RMC because inaccurate estimates of the  $Q$ -value lead to incorrect control decisions. One should note that in Algorithm 4.2, at time  $k = 0$ , running the forward simulations to generate samples of the pathwise cost requires  $K$  sequential samples of the state transition probability for each particle. Moreover, as  $k$  decreases in Algorithm 4.1, the variance of the pathwise costs increases, and accordingly, robust / regularized regression will be required to mitigate these effects.

A number of solutions are available to deal with the said challenge, and include such techniques as radial basis functions, smoothing splines, neural networks, multivariate adaptive regression splines (MARS),  $\ell_1$ -regularized regression, random forests, and others. Each approach has its own tradeoffs in regard to tuning, computational requirements, scalability (both in the number of observations and dimensions), and predictive power, and we refer the reader to [16] for a detailed overview of each approach.

One particularly effective approach that we adopt here (with a slight modification) was presented in [5] and involves building a  $k$ -d tree for the initial

condition set  $Z$  and applying linear regression at each leaf. In particular, one takes  $L_j$  partitions of the  $j^{\text{th}}$  component of the state vector, such that there are a total of  $N_p = \prod_j L_j$  partitions of the state space, each having the same number of particles. One should also take care to bound the domain of the local functions at each leaf for the purposes of extrapolation, which we discuss later. This entire process is illustrated in Figure 4.2 with a 2-dimensional example, where the bounds of each domain, denoted by  $D_\ell$  for  $\ell \in \{1, 2, \dots, N_p\}$ , are determined by the outmost points along each direction. In the presence of outliers, one may also wish to either remove the outliers before setting the domain limits or limit the domain to a fixed number of standard deviations along each direction, where standard deviation is computed using only the particles at the given leaf. We employ the former practice, which is illustrated in Figure 4.2. The overall domain is  $\mathcal{D} := \bigcup_\ell D_\ell$ .

The algorithm scales well with the number of sample points  $M$ , which we take to be a multiple of  $N_p$ . However, it is exponential in the dimension  $n$  and is hence suited to moderate dimensional problems. Another aspect of the original algorithm is that it fits linear models at each leaf using standard least squares; consequently, the fits are not explicitly robust to the high-variance samples and cannot capture nonlinearities in the response surface. To address these limitations, we propose using an  $\ell_1$ -regularized quadratic fit in each partition.



**Figure 4.2:** Partitioning scheme for  $L = (2, 2)$  and  $\zeta^{(i)} \sim \mathcal{N}(0, C)$  with  $C \in \mathbb{R}^{2 \times 2}$ . Here,  $i \in \{1, 2, \dots, M\}$  with  $M = 1,200$ ,  $c_{1,1} = c_{2,2} = 10$ , and  $c_{2,1} = c_{1,2} = 2$ . The black rectangles indicate the individual domains. The split in the  $\zeta_1$  coordinate happens roughly at zero to divide the number of particles in half, while the splits in the  $\zeta_2$  coordinate further subdivide the particles such that each bin contains 300 points.

Let  $\mathcal{I}_\ell \subset \{1, 2, \dots, M\}$  denote the subset of the indices of the particles that belong to partition  $\ell$ , with  $|\mathcal{I}_\ell| = m = M/N_p$ . Furthermore, with  $\mathcal{I}_\ell = \{i_1, i_2, \dots, i_m\}$ , we take  $y^{(\ell)} := (q_{i_1}, q_{i_2}, \dots, q_{i_m}) \in \mathbb{R}^m$ . Here,  $q_i$  is the path-

wise cost sample that is generated from Algorithm 4.2 and associated with data site  $\zeta^{(i)} \in Z$ , where  $i \in \mathcal{I}_\ell$  and the initial condition set  $Z$  takes the form  $Z = \{\zeta^{(1)}, \zeta^{(2)}, \dots, \zeta^{(M)}\}$ . Thus  $y^{(\ell)} \in \mathbb{R}^m$  denotes the local response vector for the data sites belonging to partition  $\ell$ , i.e., for data sites  $\zeta^{(i)} \in Z$  such that  $i \in \mathcal{I}_\ell$ . Additionally, we denote by  $H^{(\ell)} \in \mathbb{R}^{m \times N_b}$  the predictor matrix, where  $N_b = n + n(n + 1)$  is the number of basis functions, not including the constant term. Thus, the rows of the predictor matrix take the form  $H_{i*}^{(\ell)} = h^\top(\zeta^{(i)})$ , where  $h : \mathbb{R}^n \rightarrow \mathbb{R}^{N_b}$  is used to evaluate the quadratic basis functions for each data site  $\zeta^{(i)}$  in partition  $\ell$ . We assume that the responses take the following form

$$y^{(\ell)} = H^{(\ell)}\beta^{(\ell)} + \beta_0^{(\ell)}\mathbf{1}_{m \times 1} + \epsilon^{(\ell)}. \quad (4.13)$$

Here  $\epsilon^{(\ell)} \in \mathbb{R}^m$  is the vector of residuals in partition  $\ell$ ,  $\mathbf{1}_{m \times 1}$  is an  $m$ -length vector of all ones, and  $\beta_0^{(\ell)} \in \mathbb{R}$  and  $\beta^{(\ell)} = (\beta_1^{(\ell)}, \beta_2^{(\ell)}, \dots, \beta_{N_b}^{(\ell)}) \in \mathbb{R}^{N_b}$  are the coefficients to be determined in the regression. To minimize the residuals in a robust fashion, we minimize

$$\|y^{(\ell)} - \beta_0^{(\ell)}\mathbf{1}_{m \times 1} - H^{(\ell)}\beta^{(\ell)}\|_2 + \lambda\|\beta^{(\ell)}\|_1, \quad (4.14)$$

where  $\lambda > 0$  is a tuning parameter. As noted in §3.4.4 of [16], this problem can be solved in the same time complexity as regular least squares, and hence



it is suited for repeated use in the partitioned regression scheme, i.e., for each  $\ell \in \{1, 2, \dots, N_p\}$ .

Overall, if the total response vector  $q \in \mathbb{R}^M$  was generated at time  $k$  for control action  $u$ , then the estimator for the  $Q$ -value in Algorithm 4.1, takes the form

$$\hat{Q}_k(z, u) = \langle \beta^{(\ell)}, h(z) \rangle + \beta_0^{(\ell)}, \quad z \in \mathcal{D}_\ell \quad (4.15)$$

where  $\langle \cdot, \cdot \rangle$  denotes inner product and  $h(z) \in \mathbb{R}^{N_b}$  is the aforementioned mapping that formed the rows of the predictor matrix  $H^{(\ell)}$ . Thus, to evaluate the  $Q$ -value estimate for a given  $z \in \mathcal{Z}$ , one must first determine the particular partition to which the state belongs.

Ideally, the controller keeps the steady state trajectories in the domain  $\mathcal{D}$ ; however, in the case that extrapolation must be performed, one should be wary of the behavior of the quadratic fit outside  $\mathcal{D}$ . Thus, for  $z \in \mathcal{Z} \setminus \mathcal{D}$ , we determine

$$z_e = \arg \min_{s \in \mathcal{D}} \|z - s\|_1 \quad (4.16)$$

and evaluate the fit at  $z_e$ , which we have found to produce satisfactory performance. Moreover, we expect the partitioned quadratic fits to interpolate well in the domain but avoid their use for extrapolation.

## 4.3 Regression Monte Carlo for Target Tracking

We now present the modified regression Monte Carlo algorithm in the context of vision-based target tracking. In particular, we first present the adaptation to Algorithm 4.1 that addresses the partially discrete state space described at the end of Section 2.2.1. Secondly, in light of the smaller state space one needs to consider in creating the optimal policy maps, we describe the initial condition set  $Z$  and the corresponding partitioning scheme used in the regression.

### 4.3.1 Modified Algorithm

While the stochastic kinematic model for the UAV captures realistic dynamics encountered with a small fixed wing aircraft, it also yields a partially discrete state space. As a result, the standard RMC algorithm must be slightly modified for the application of vision-based target tracking with two UAVs. In particular, we will need to run Monte Carlos simulations for roll-angle pairs  $r \in \mathcal{C} \subseteq C^2$  combined with all allowable roll action pairs  $u \in U(r)$ , where  $\mathcal{U}(r)$  is given by (4.1).

Note that due to symmetry arguments discussed later, one need not consider all roll-action pairs, which entails that  $\mathcal{C} \subset C^2$ . To accommodate these modifications, we begin by defining the continuous state space to be  $\mathcal{X}$ , which has

dimension  $n_r = 7$  and is simply  $\mathcal{Z}$  with the discrete roll states  $r \in \mathcal{C}^2$  removed. Thus, with the state descriptions of Section 2.2.1 and the individual UAV poses  $\mathbf{p}_j$  described in Section 4.1.2, the state vector  $\chi \in \mathcal{X} \subset \mathbb{R}^7$  is given by

$$\chi := (\mathbf{p}_1, \mathbf{p}_2, v). \quad (4.17)$$

Lastly, the initial condition set described in Section 4.2.2 is denoted by  $X$ .

The modified RMC algorithm for vision-based target tracking with two-UAVs is presented in Algorithm 4.3. The primary differences with Algorithm 4.1 is the addition of a **for** loop around the discrete-valued states, as well the formation of the full initial condition set  $Z$  from the set  $X$  of continuous initial condition states and the given roll-angle pair  $r \in \mathcal{C}$ . Furthermore, the regression is performed using only the continuous states in  $X$ , and thus the dimensionality of the regression problem is reduced to  $n_r = 7$ . In practice, the two innermost loops are often combined and run in parallel for increased computational performance. On a final note, when generating the cumulative cost samples for each roll action with Algorithm 4.2, one should replace  $U$  with  $\mathcal{U}(r)$  in the requirements section and  $U$  with  $\mathcal{U}(r^{(i)})$  in Line 8.

---

**Algorithm 4.3** Regression Monte Carlo for Target Tracking

---

**Require:** Initial condition set  $X \subset \mathbb{R}^7$ , where  $|X| = M$ ; set of roll-angle pairs  $\mathcal{C} \subseteq C^2$ ; action space  $\mathcal{U}(r)$

```

1:  $N_r \leftarrow |\mathcal{C}|$ 
2: for  $k = K - 1, K - 2, \dots, 0$  do
3:   for  $s = 1, 2, \dots, N_r$  do
4:     Form initial condition set  $Z$  from  $X$ , such that for  $\zeta^{(i)} \in Z$  and
        $\chi^{(i)} \in X$ , the following component-wise relationships hold:
        $\zeta_{(4j-3):(4j)}^{(i)} = (\chi_{(3j-2):(3j)}^{(i)}, r_j^{(s)})$  and  $\zeta_9^{(i)} = \chi_7^{(i)}$ , where  $j \in \{1, 2\}$  and
        $i \in \{1, 2, \dots, M\}$ 
5:      $N_u \leftarrow |\mathcal{U}(r^{(s)})|$ 
6:     for  $\ell = 1, 2, \dots, N_u$  do
7:       Using Algorithm 4.2, generate cumulative cost realization vector
        $q \in \mathbb{R}^M$  corresponding to control action  $u^{(\ell)} \in \mathcal{U}(r^{(s)})$ 
8:       Regress  $q_i$ 's against statistics derived from corresponding  $\chi^{(i)}$ 's
       to determine  $\hat{Q}_k(z, u^{(\ell)})$ 
9:     end for
10:  end for
11: end for
12: return  $Q$ -value approximators  $\hat{Q}_k(z, u)$ , where  $k \in \{0, 1, \dots, K - 1\}$ 

```

---

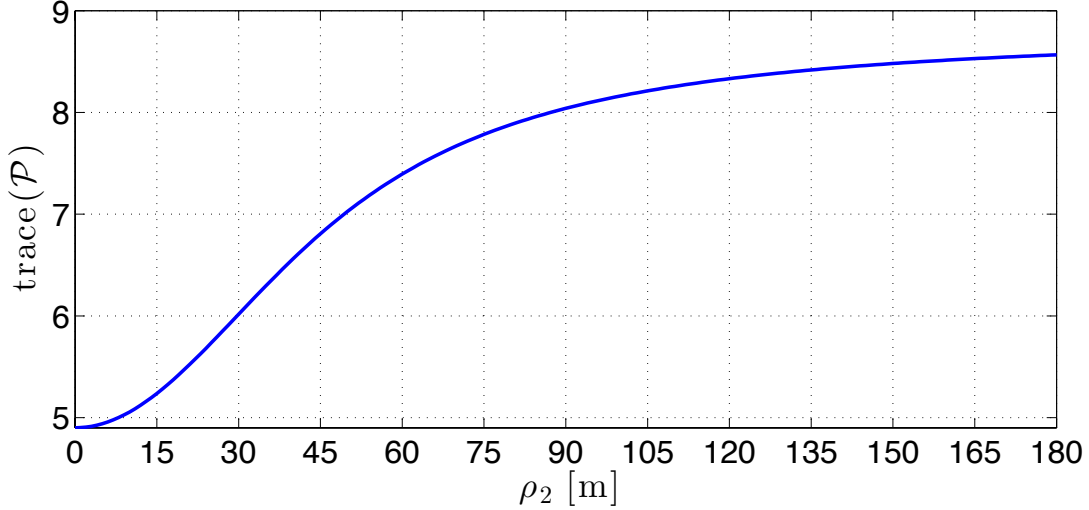
### 4.3.2 Initial Condition Set

While modified RMC approach offers significant computational savings over a basic Monte Carlo method for value iteration, it generally requires adjusting the initial condition set and the regression to obtain satisfactory performance. Thus, we now describe the initial condition set  $X$  of Algorithm 4.3 that comprises the continuous states. For UAV  $j$ , if the relative position  $\mathbf{r}_j$  is represented in polar coordinates as  $(\rho_j \cos \vartheta_j, \rho_j \sin \vartheta_j)$ , then we take  $\rho_j$  to be normally distributed with mean  $\mu_\rho > 0$  and variance  $\sigma_\rho^2$  while  $\vartheta_j$  is uniformly distributed on  $[\underline{\vartheta}, \bar{\vartheta}]$ .

Typically,  $(\underline{\vartheta}, \bar{\vartheta}) = (-\pi, \pi)$ . However, if one exploits symmetry, this need not be the case. Also, in the process of generating  $M$  samples of  $\rho_j$ , we only retain those samples that have strictly positive values and those that are within  $3\sigma$  of the mean, as the the outer boundaries of the partitioning domain are set in the manner discussed and illustrated in Section 4.2.3. Next, we take  $\psi_{r,j}$ , the relative heading angle of UAV  $j$ , to be uniformly distributed on  $[-\pi, \pi]$  and the target speed  $v$  to be uniformly distributed on  $[\underline{v}, \bar{v}]$ , where  $\underline{v}$  and  $\bar{v}$  are discussed in Section 4.1.1. Moreover, at the start of Algorithm 4.3, we generate  $M$  samples of the continuous states in the manner just described to form  $X$ , where the mean  $\mu_\rho$  and variance  $\sigma_\rho^2$  of the radial distribution of the relative planar position states are tuning parameters set beforehand.

### 4.3.3 Barrier Function

While in the single UAV case, the stage cost given by (3.28) directly penalizes distance from the target, this is not the case for both agents in the two-UAV scenario. In particular, having UAV 1 on top of the target and UAV 2 far away is only slightly worse than having both on top of the target. This is illustrated in Figure 4.3 and suggests that the cost function is not radially unbounded with respect to the second UAV's planar distance from the target. One should take



**Figure 4.3:** Cost function  $g(z) = \text{trace}(\mathcal{P})$  with the target located at the origin and the first UAV located on top of the target at an altitude of 40 [m]. Note that the separation angle  $\gamma$  is 0 since the first UAV's planar distance is  $\rho_1 = 0$ ; consequently, the fused GEC is completely characterized by the second UAV's planar distance  $\rho_2$  from the target. The second UAV has an altitude of 45 [m].

note of the range of values for the cost function in this scenario versus that of Figure 3.4.

As prediction accuracy of the  $Q$ -value functions is highest where many simulation scenarios reside, we wish to ensure that both UAV's remain close. This is facilitated through the addition of a barrier-type function to the stage cost that is non-negative, radially unbounded, and only nonzero for large distances. Hence, we present the following augmented cost function to be used in the dynamical optimization of Algorithm 4.3:

$$g_b(z) := \text{trace}(\mathcal{P}) + b(z), \quad (4.18)$$

where

$$b(z) = \sum_{j=1}^2 \max\{0, \rho_j - (\mu_\rho + 2\sigma_\rho)\}, \quad (4.19)$$

$\rho_j$  denotes UAV  $j$ 's planar distance from the target, and  $\mu_\rho$  and  $\sigma_\rho$  are the mean and standard deviation of the normally distributed planar distances from the target that form the initial condition set described in Section 4.3.2. Moreover, the barrier function  $b(z)$  is zero along ideal trajectories and ensures all UAVs remain within a reasonable distance of the target.

## 4.4 Results

We now demonstrate the nature of the optimal coordination strategy and the effectiveness of the modified RMC approach in the present application of optimally coordinating two UAVs to perform vision-based target tracking in a stochastic environment. To establish the benefit of the proposed control approach, we compare the performance of the (approximately) optimal controller against an effective baseline strategy, as well as the traditional approach of coordinated standoff tracking. Finally, we analyze the nature of the optimal control policy to determine the predominant behavior of the two fourth-order UAVs as they cooperatively track the stochastic ground target.

**Table 4.2:** General Parameters

Parameter	Description	Value	Units
$R_{\tilde{\theta}}$	sensor attitude covariance	$9I_{3 \times 3}$	$\text{deg}^2$
$h_a$	UAV altitudes	$\{40, 45\}$	m
$T_s$	zero-order hold period	2	s
$K$	planning horizon	15	-

**Table 4.3:** RMC Parameters

Parameter	Description	Value
$m$	particles per partition	10,000
$L$	partitioning scheme	$(2, 2, 4, 2, 2, 4, 2)$
$\lambda$	regularization parameter	3
$\mu_\rho$	radial distribution mean	70
$\sigma_\rho^2$	radial distribution variance	$35^2$

#### 4.4.1 Problem Setup and Solution Parameters

Throughout this section we extensively analyze the results of a fairly realistic tracking scenario that is summarized by the parameters pertaining to the UAVs and target provided in Tables 2.5 and 4.1, respectively. The scenario is similar to that considered in [7], wherein Collins et al. presented field test results for a single UAV (capable of  $15 - 20$  [m/s] airspeeds) tracking a target that traveled between  $5 - 10$  [m/s]. One will immediately notice that we do not consider stopped targets nor target speeds approaching the UAVs' airspeeds, as we wish to focus on an interesting regime of the relative UAV-target dynamics.



The parameters pertaining to the general dynamical optimization of Section 4.1.3 are presented in Table 4.2. The planning horizon of  $KT_s = 30$  seconds was chosen to be longer than the time it takes a UAV to perform a loop at max bank, which for maximum turn rate of  $\omega_{\max} = \alpha_g \tan(2\Delta)/s_a$  [rad./s] is approximately 20 seconds. This time was chosen such that the UAVs will always realize the long-term impact of committing to a loop, as the control policy  $\mu_k^*(z)$  is applied in a receding horizon fashion, i.e., we always apply  $\mu_0^*(z)$  at every time step.

The parameters pertaining to the RMC solution are presented in Table 4.3, where the cost function from (4.18) was minimized using Algorithm 4.3 and the techniques for computational savings presented in Appendix A. To be clear, throughout this section the terms *cost* and *stage cost* refer to the original cost function given by (3.28), which is simply the trace of the fused GEC. Also, we henceforth refer to the resulting policy as the *optimal* policy with the understanding that this policy is in reality an approximation to the true optimal policy.

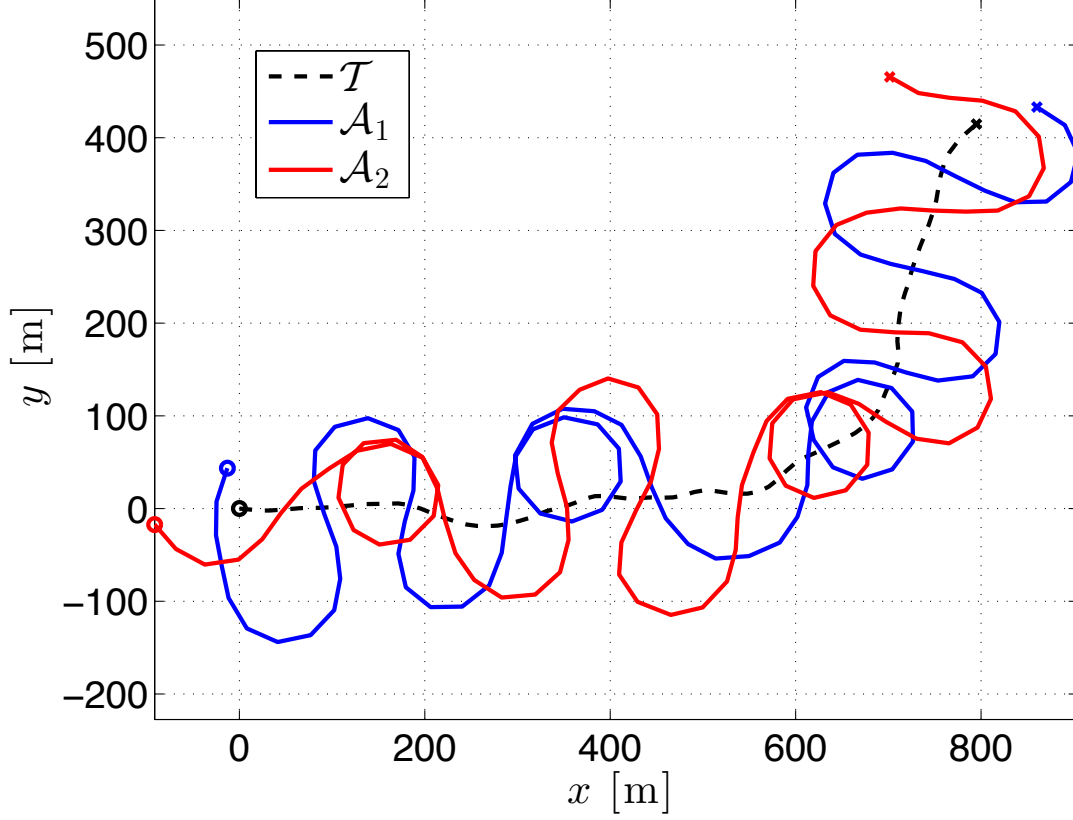
Regarding the regression, each partition had 36 degrees of freedom in the quadratic regression, as the dimension of the continuous state space  $\mathcal{X}$  is 7. Hence, we chose the particular values of  $m$  to avoid overfitting while  $\lambda$  was chosen to add robustness to process noise, where  $\lambda \in [3, 10]$  generally works well.

Regarding the partitioning scheme, recall that  $L \in \mathbb{N}^7$  denotes the number of partitions of each component of the continuous state space  $\mathcal{X}$ , and hence the total number of partitions is  $N_p = \prod_j L_j = 512$ , though one does not need to estimate the  $Q$ -value in more than 320 partitions according to the symmetry considerations of Appendix A. Note that choosing 2 partitions for the relative  $x$  and  $y$  coordinates implies that the partitions of the planar positions correspond (approximately) to standard Cartesian quadrants, since the 2-dimensional distributions that generate the individual (planar) position samples of the initial condition set are radially symmetric per the discussion of Section 4.3.2. The mean and variance of the normally distributed planar distances are also given in this table. Also, we have found that the relative heading coordinates are the most sensitive to the number of regression partitions (due to the nonlinearity) and hence choosing  $L_3$  or  $L_6$  to be less than 3 typically yields poor performance. Overall, through considerable testing, this particular partition configuration is a good compromise between computational feasibility and mitigating the effects of the nonlinearity through additional partitions.

To highlight key features of the optimal trajectories, we have provided a representative sample trajectory in Figure 4.4 and its corresponding performance in Figure 4.5, where the initial condition was arbitrarily chosen. From Figure 4.4, one should note how the optimal trajectories comprise both sinusoidal and orbital

trajectories, where the latter is not necessarily centered around the target. At the beginning of the simulation, one UAV is performing an “S” turn (sinusoidal pattern) while the other is performing a loop. The UAVs switch roles and perform the same joint maneuver before both UAVs make out-of-phase loops and then out-of-phase “S” turns. From Figure 4.5, distance coordination becomes apparent, as the peaks of the distance curves alternate. The second subplot of this figure indicates that the UAVs do not strive to maintain orthogonal viewing angles, as the curve does not oscillate around  $\gamma = 90^\circ$ . However, the UAVs do benefit from orthogonal viewing angles when they are both moderately far, e.g.,  $t = 82$  [s], where the cost is kept from spiking by such a configuration. Overall, minimum distance is the dominant factor in the cost function, though viewing angle coordination does benefit the UAVs when they find themselves moderately far from the target.

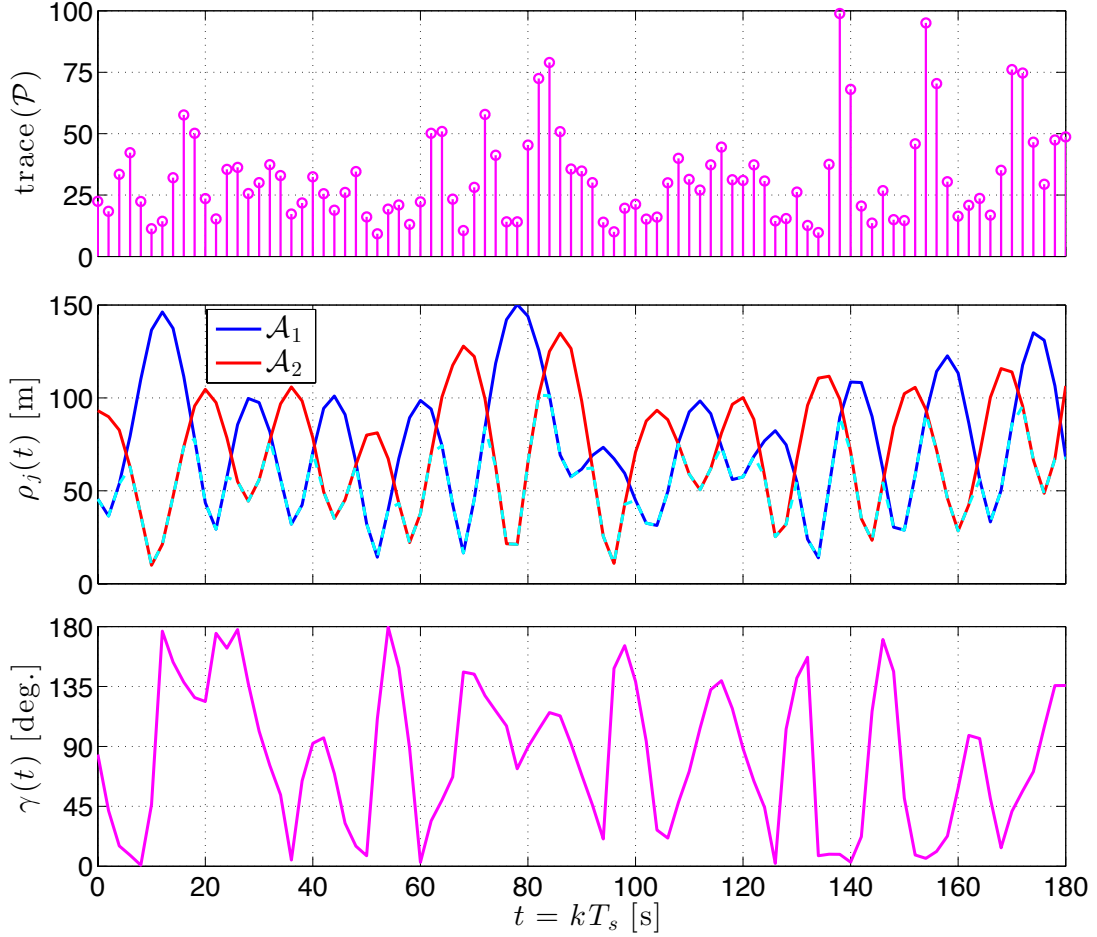
While this single realization of a controlled stochastic process does not establish distance coordination as the predominant coordination strategy, it does illustrate typical behaviors encountered with this policy. Namely, the optimal trajectories comprise a rich mixture of sinusoidal and orbital trajectories that occasionally pass over or near the target rather than just a single trajectory type, which is the primary goal in the vast majority of the target tracking literature.



**Figure 4.4:** Optimally coordinated trajectories over a three minute window. The starting positions of all vehicles are marked by an “o” while the ending positions are denoted by an “x”. The target (denoted by  $\mathcal{T}$ ) begins at the origin travelling at approximately 5.4 [m/s] and finishes its trajectory travelling at approximately 7.3 [m/s]. Both UAVs (denoted by  $\mathcal{A}_1$  and  $\mathcal{A}_2$ ) begin with zero roll.

#### 4.4.2 RMC Performance

Since RMC is an approximate solution to the problem described in Section 4.1.3, we must first compare the resulting policy against alternative control strategies to warrant the considerable effort. Unfortunately, alternative strategies will not provide a suitable comparison, as the analytical methods of Sec-



**Figure 4.5:** Performance metrics of optimally coordinated UAVs: planar distances  $\rho_j$ , separation angle  $\gamma$ , and trace of the fused GEC  $\mathcal{P}$ . The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

tion 3.1.1 were proposed for different problem formulations. In particular, the strategies utilizing nonlinear feedback or vector fields assume either simplified third-order UAV kinematics, an additional input of airspeed control, or (piece-wise) constant target velocity. Moreover, even if we implement these control

approaches with simplified UAV dynamics under a constant airspeed assumption, their robustness to a target that randomly turns and accelerates is not known. Nonetheless, we shall later make a brief comparison with a coordinated standoff controller at the minimum allowable standoff distance (determined by the maximum bank angle) in an ideal setting. Regarding the optimization-based approaches for two or more UAVs, these strategies optimize the given cost function for simplified UAV kinematics with a short planning horizon, which is not suitable for the present problem formulation. Moreover, a comparison of the present approach with any of the works of Section 3.1.1 would be not be fair.

### **Comparison with Uncoordinated Optimal Controllers**

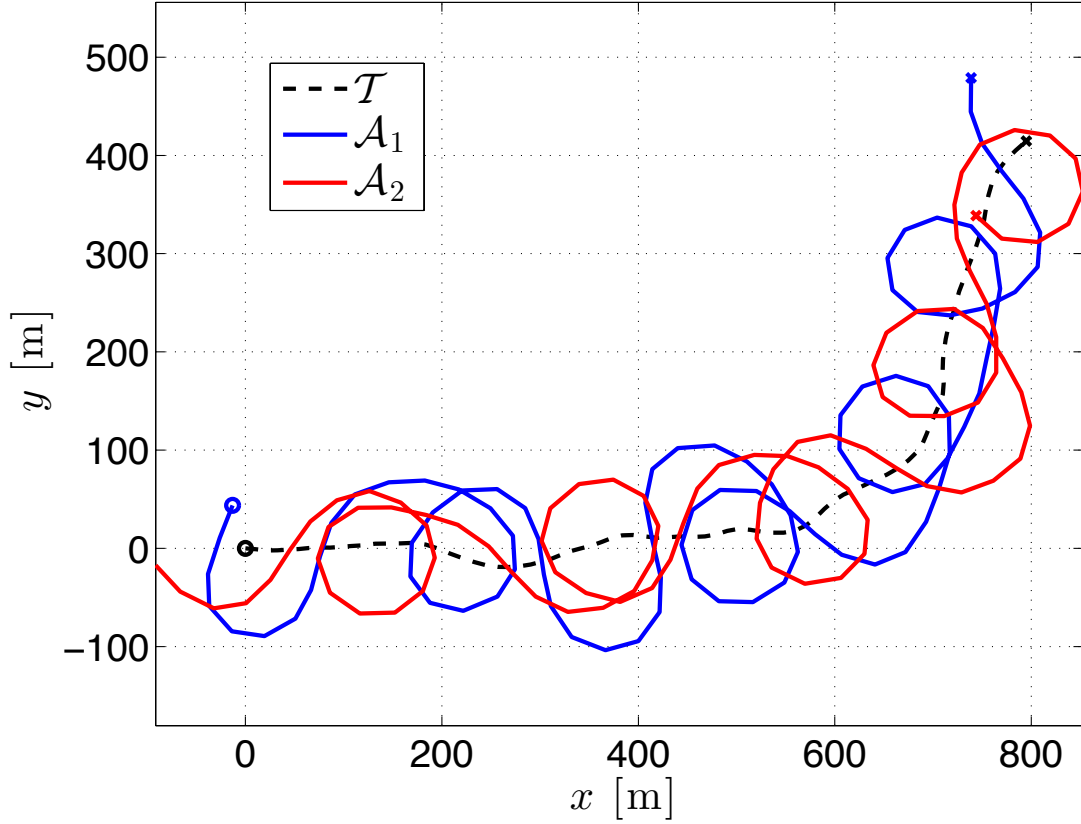
To generate an appropriate baseline strategy, we solve the stochastic optimal control problem of Section 4.1.3, but for a single UAV, as this essentially amounts to having each UAV minimize its own individual squared distance from the target. Recall that distance is the dominant factor in the fused GEC, and hence this is a both a reasonable and effective approach. Since the problem for a single UAV has modest dimension, one can solve it using the basic Monte Carlo solution of Section 2.3.4. As a result, we performed value iteration according to Section 2.3.4 to generate two individual control policies with the cost function (3.28) and the parameters of Table 4.2. We used  $M = 1,000$  Monte Carlo

**Table 4.4:** State Space Discretization in One-UAV Scenario

Set	Description	Value	Units
$Y$	relative positions	$\{-225, -220, \dots, 225\}$	m
$\Psi$	relative headings	$\{0, 15, \dots, 345\}$	deg.
$C$	roll commands	$\{0, \pm 15, \pm 30\}$	deg.
$W$	target speeds	$\{4.5, 5.0, \dots, 12.5\}$	m/s
$Z$	discrete state space	$Y^2 \times \Psi \times C \times W$	-

samples in (3.33) with a finite state space  $Z$  described by Table 4.4. We denote the resulting policies as  $\pi_k^{(1)}(\zeta_k^{(1)})$  and  $\pi_k^{(2)}(\zeta_k^{(2)})$ , where  $k \in \{0, 1, \dots, K-1\}$  and  $\zeta^{(1)}$  and  $\zeta^{(2)}$  are the 5-dimensional states of the individual UAVs whose components are described at the end of Section 2.2.1. Note that, as in the case of coordinated UAVs, we always apply these policies in a receding-horizon fashion, i.e., we always use the time-stationary policies  $\pi_0^{(1)}(\zeta_k^{(1)})$  and  $\pi_0^{(2)}(\zeta_k^{(2)})$  for the uncoordinated UAVs for all  $k \in \mathbb{Z}_{\geq 0}$ .

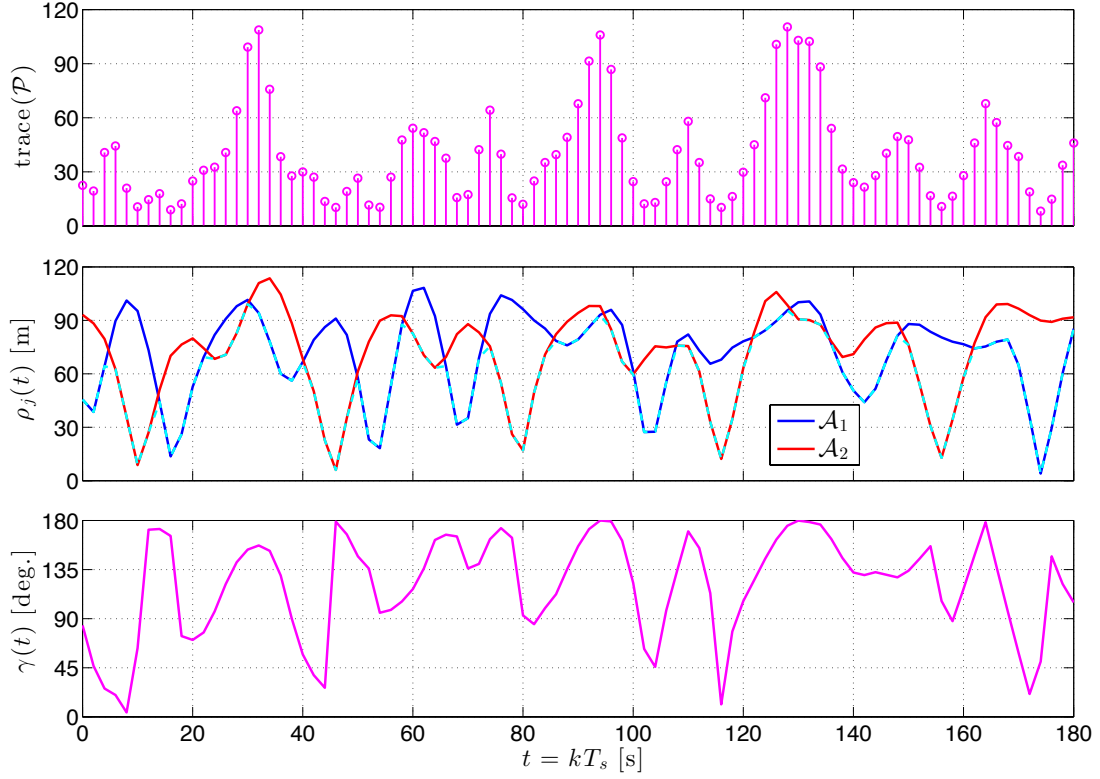
To illustrate the nature of this control strategy, we have provided plots in Figures 4.6 and Figures 4.7 illustrating the behavior and performance of uncoordinated controllers for the same initial conditions and target trajectory realization as in Figures 4.4 and 4.5. Note that while each UAV minimized its own individual GEC in the dynamic optimization, it is the fused covariance that is plotted in the top subplot of Figure 4.7. Perhaps the most noticeable and distinguishing feature of Figure 4.6 is the fact that the UAVs primarily make orbital trajectories around the target, which enables them to keep their worst-



**Figure 4.6:** Uncoordinated trajectories over a three minute window. The starting positions of all vehicles are marked by an “o” while the ending positions are denoted by an “x”. The remaining notation, initial conditions, and target trajectory are the same as in Figure 4.4.

case distance from the target smaller. This can be seen in the top subplot of Figure 4.7, where the peak planar distance from the target is approximately 114 [m], whereas that of Figure 4.5 is approximately 150 [m]. One can also see the lack of coordination for  $t \in [126, 132]$ , as the cost is above 100 [m<sup>2</sup>] during this time period when both UAVs are moderately far from the target and have viewing angles that are quite far from being orthogonal. The time-averaged cost





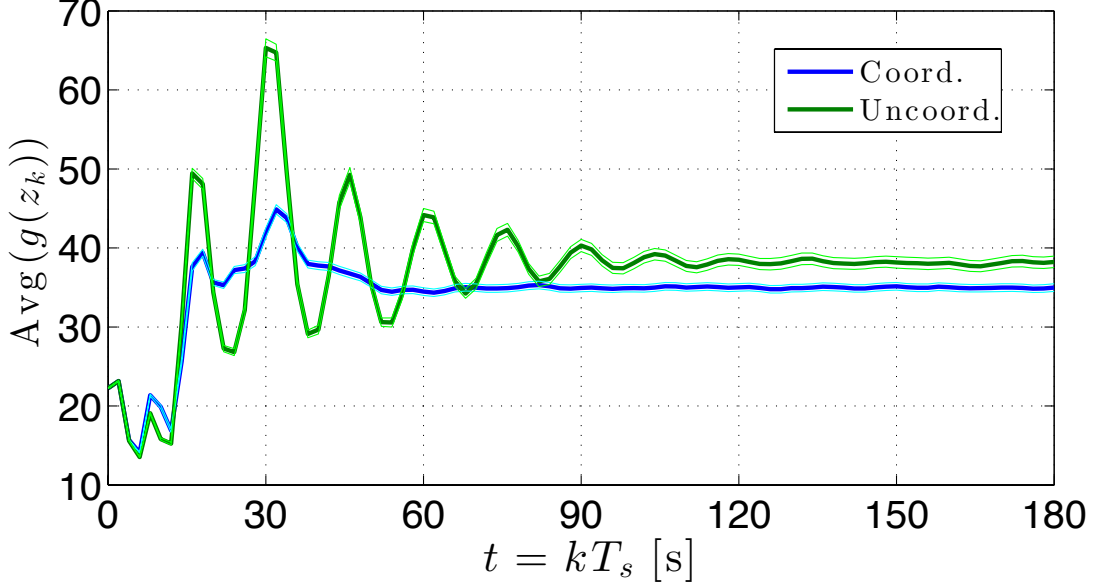
**Figure 4.7:** Performance metrics of uncoordinated UAVs: planar distances  $\rho_j$ , separation angle  $\gamma$ , and trace of the fused GEC  $\mathcal{P}$ . The minimum distance  $\varrho := \min\{\rho_1, \rho_2\}$  is indicated by the dashed (cyan) line in the second subplot.

for this run was approximately  $39.6 \text{ [m}^2\text{]}$  while that of the coordinated control policy was  $32.4 \text{ [m}^2\text{]}$ . Thus, the optimal trajectories are rather counterintuitive in that coordination allows the UAVs to deviate further from the target without sacrificing performance, which is a valuable feature. Of course, this deviation must be done in a alternating fashion, as illustrated by the distance curves of Figure 4.4.

To better demonstrate the temporal nature of both control strategies in an expected sense, we have selected an initial condition that is a good starting point for both strategies and run 50,000 Monte Carlo simulations from this initial condition with the same realizations of target trajectories to compute both the mean value and 98<sup>th</sup>-percentile statistics of the cost, which are provided in Figures 4.8 and 4.9, respectively. By inspecting Figure 4.8, one can see that the optimal control policy converges to the mean steady-state cost of (approximately) 35 [m<sup>2</sup>] within one minute while the uncoordinated controllers take nearly 2 minutes to converge to the mean steady-state cost of (approximately) 38 [m<sup>2</sup>]. In addition, the peak average value is significantly less in the case of the optimal control policy than in the case of uncoordinated policies. Note that the distribution of steady-state costs is independent of the initial conditions. Moreover, from a dynamical systems standpoint, the response of the uncoordinated control policies resembles the step response of an oscillatory but inherently stable plant with no controller while that of the optimal control policy resembles the step response of the same plant with a fairly well-tuned controller in the feedback loop. This results from the optimally coordinated policy minimizing the fused GEC directly while the uncoordinated policies minimize each agent's own individual GEC, which only has an indirect effect on the fused GEC.

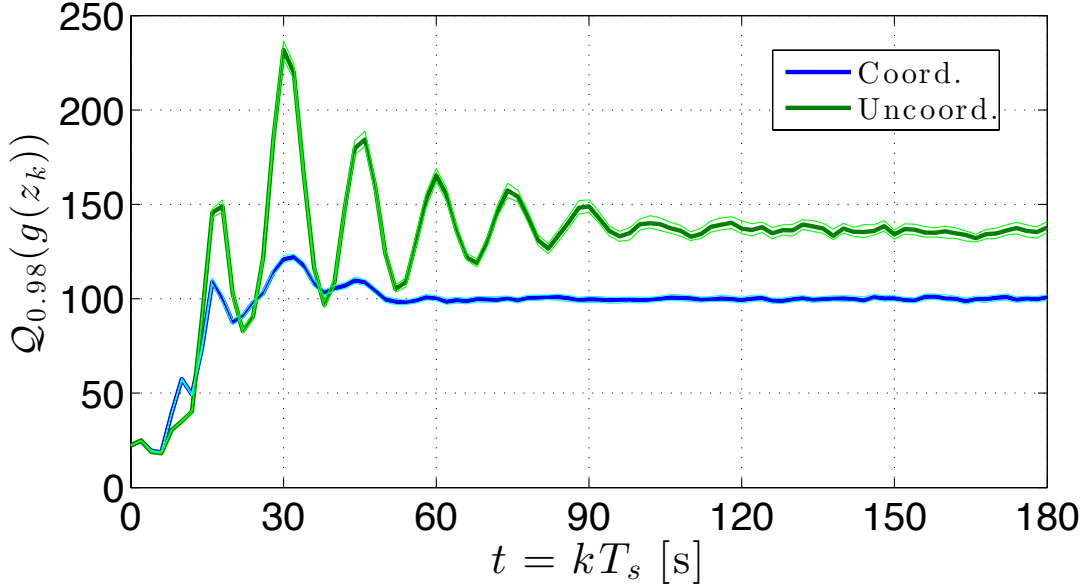
Another benefit of the coordinated control policy can be seen in Figure 4.9, where the plot indicates that the tail of the steady-state cost distribution is often significantly wider in the case of uncoordinated policies. In fact, the 98<sup>th</sup>-percentile of the steady-state costs for the uncoordinated policies is about 33% higher than that of the optimal policy. Moreover, although we have illustrated transient response performances for a specific initial condition, the plots in Figures 4.8 and 4.9 illustrate typical benefits of the optimal control policy. Namely, the recovery from initial conditions is typically faster (in an expected sense), and the tail of the cost distribution is significantly smaller in steady state, which entails that high cost events are more rare than in the uncoordinated case.

To provide a more objective comparison, we have performed another test that is independent of initial conditions. More specifically, we generated  $M = 50,000$  initial conditions randomly according to Section 4.3.2 and then ran 12-minute Monte Carlo simulations with each control strategy from these initial conditions using the same realizations of target trajectories for each approach. To reduce the effects of initial conditions, we truncated the first two minutes of each run. Computing the sample mean (over time) of the stage costs associated with each run yields the histogram presented in Figure 4.10, where the sample means are denoted by  $\bar{g}^{(i)}$ ,  $i \in \{1, 2, \dots, M\}$  and are given by  $\bar{g}^{(i)} = (1/301) \sum_{k=60}^{360} g(z_k^{(i)})$  with  $g(\cdot)$  from (3.28). Hence, whereas the previous test illustrated the first few



**Figure 4.8:** Transient response of mean value for initial condition  $z_0 = (-60, 0, -\pi/2, -30^\circ, 0, -60, 0, -30^\circ, 8.5)$ . In this initial condition, the UAVs have orthogonal viewing angles and are banked max left at a distance roughly equal to their minimum turning radius of 57.2 [m]. The lighter, thinner lines indicate the 95% confidence intervals of the mean value.

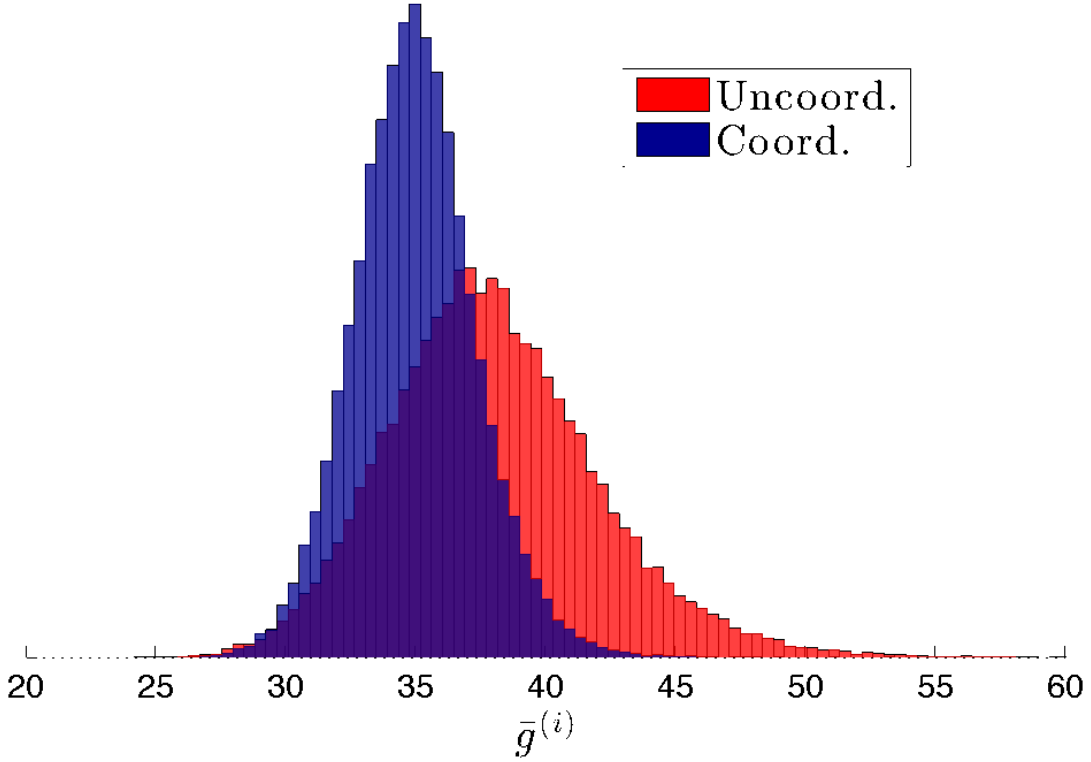
minutes of a transient response and computed certain statistics across samples, here we are computing the mean over time with the first few minutes of each simulation removed. In this plot, the sample mean and sample standard deviation of the time-averaged costs associated with the optimal policy are 34.93 [m<sup>2</sup>] and 2.33 [m<sup>4</sup>], respectively; those associated with the uncoordinated control policies are 37.95 [m<sup>2</sup>] and 4.07 [m<sup>4</sup>], respectively. Furthermore, the standard error of the mean is less than 0.02 in both cases. Moreover, while the optimally coordinated control policy reduces the mean of the time-averaged costs by only about 8%,



**Figure 4.9:** Transient response of 98<sup>th</sup> percentile for initial condition  $z_0 = (-60, 0, -\pi/2, -30^\circ, 0, -60, 0, -30^\circ, 8.5)$ . In this initial condition, the UAVs have orthogonal viewing angles and are banked max left at a distance roughly equal to their minimum turning radius of 57.2 [m]. The lighter, thinner lines indicate the 95% confidence intervals for the 98<sup>th</sup> percentile .

it reduces their standard deviation by nearly 43%. This reduction in standard deviation is illustrated by the widths of the distributions, which is considerably less in the case of the optimally coordinated strategy. Thus, the optimally coordinated control policy improves the predictability of the tracking performance substantially.

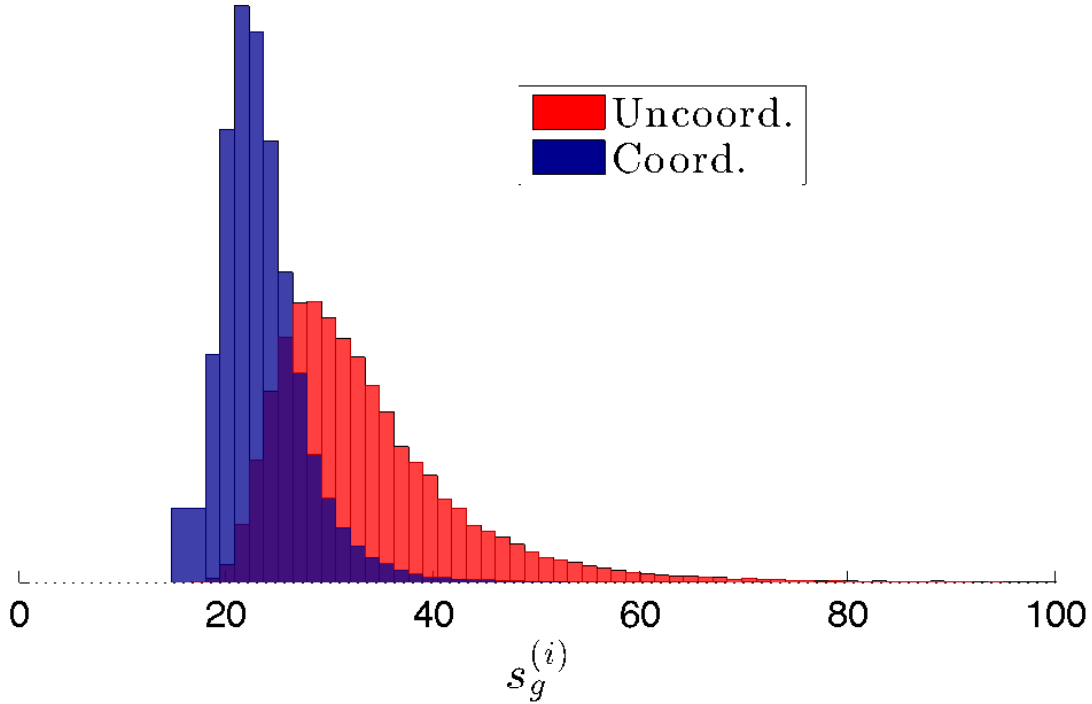
Another comparison is the fluctuation of the cost about the mean value for each of the preceding Monte Carlo runs, which is given by the sample standard deviation of the stage costs associated with each simulation. These



**Figure 4.10:** Mean value  $\bar{g}^{(i)}$  of stage costs, where  $i \in \{1, 2, \dots, M\}$ , for each of the  $M = 50,000$  ten-minute Monte Carlo simulations. Note that the outliers for the uncoordinated policy are not all shown, as they extend out to just above 66.

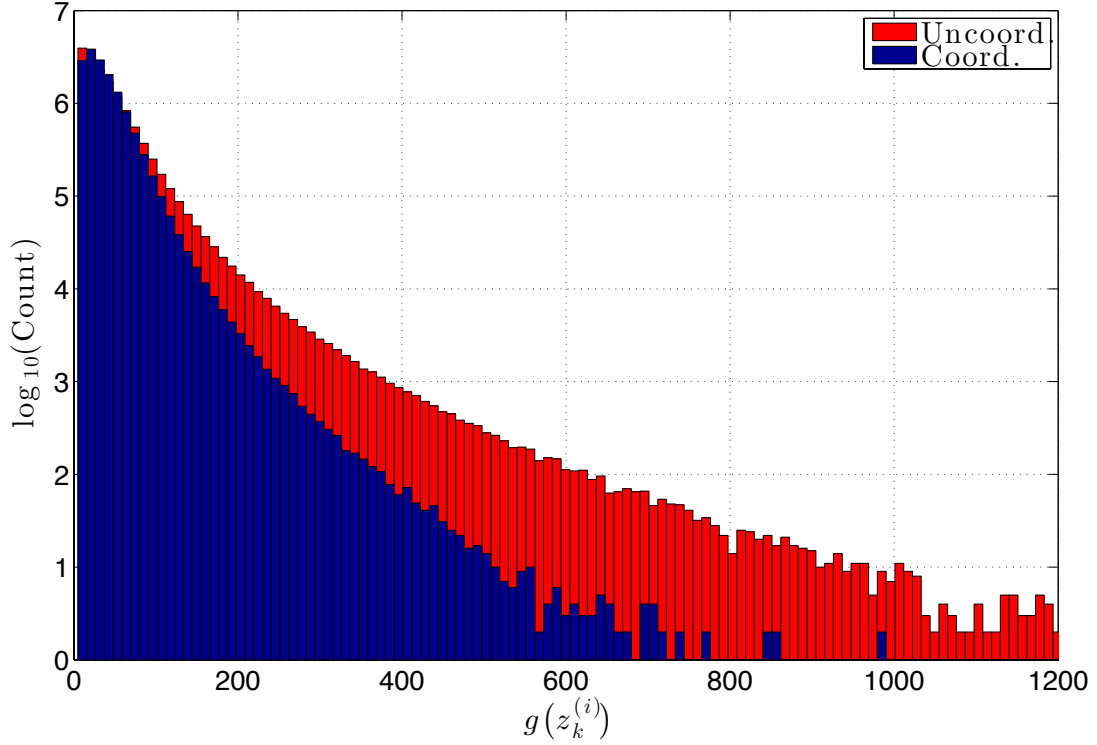
quantities, denoted by  $s_g^{(i)}$ ,  $i \in \{1, 2, \dots, M\}$ , are given by the relationship  $(s_g^{(i)})^2 = (1/300) \sum_{k=60}^{360} (g(z_k^{(i)}) - \bar{g}^{(i)})^2$  and are plotted in Figure 4.11. The histogram illustrates the fact that the optimal control policy produces fluctuations about the mean that are typically smaller and less variable across runs than those of the uncoordinated control policies.

One final comparison is provided by plotting the histogram of the steady-state costs with the effects of averaging removed. More specifically, in Figure 4.12 we



**Figure 4.11:** Standard deviation  $s_g^{(i)}$  of stage costs, where  $i \in \{1, 2, \dots, M\}$ , for each of the  $M = 50,000$  ten-minute Monte Carlo simulations. Note that the outliers for the uncoordinated policy are not all shown, as they extend to just beyond 156.

provide a histogram of the stage costs given by (3.28) at each time step for each of the  $M = 50,000$  ten-minute Monte Carlo simulations in steady state. The number of counts is presented with a logarithmic scale to focus on the tails of the distributions. One can see that the tail of the cost distribution corresponding to the uncoordinated policies decays slower than that corresponding to the optimal policy. Moreover, rare events are less frequent and less severe with the optimal control policy than with uncoordinated control policies. In fact, near  $g(z) =$



**Figure 4.12:** Stage costs during steady-state at each time step and for each of the  $M = 50,000$  ten-minute Monte Carlo simulations, i.e.,  $g(z_k^{(i)})$ ,  $\forall i \in \{1, 2, \dots, M\}$  and  $\forall k \in \{60, 61, \dots, 360\}$ . Hence, there are 15.05 million samples in each histogram.

400 [m<sup>2</sup>], the frequency of this particular cost with the optimal control policy is an order of magnitude lower than with the uncoordinated policies. Since we expect the controlled processes to be ergodic, these histograms are representative of a single very-long run for each of the cooperative tracking approaches, e.g., a run lasting hundreds of hours.



### Comparison with Standoff Tracking

To provide a brief comparison with the standoff tracking approach, we note that the minimum allowable standoff distance,  $\varrho_s$ , as imposed by the maximum bank angle  $\phi_{\max}$ , is given by Equation 5.37 in [21] as follows:

$$\varrho_s \geq \frac{(v + s_a)^2}{\alpha_g \tan(\phi_{\max})}, \quad (4.20)$$

where  $v$ ,  $s_a$ , and  $\alpha_g$  denote target speed, UAV airspeed, and gravitational acceleration, respectively. With the target traveling at the minimum allowable speed of 4.5 [m/s] and the remaining parameters given in Table 2.5, we have

$$\varrho_s \geq \frac{(4.5 + 18)^2}{9.81 \tan(30\pi/180)} \approx 89.4 \text{ [m]}. \quad (4.21)$$

In an ideal setting for standoff tracking, the target is traveling at a constant velocity and the UAVs have orthogonal viewing angles at the nominal standoff distance of  $\varrho_s = 90$  [m]. Hence, with the altitudes of Table 4.2,  $\text{trace}(\mathcal{P}) \approx 46$  [m<sup>2</sup>]. Recalling that both the time-averaged cost and ensemble-averaged cost of the optimal policy in steady state (over many target velocity realizations) were both approximately 35 [m<sup>2</sup>], one can see that the optimally coordinated policy offers a significant advantage in terms of average cost, even in this slow target scenario. If the target were instead traveling at  $v = 9$  [m/s], or half the UAV's airspeed, standoff tracking requires  $\varrho_s \geq 128.7$  [m] according to (4.20). Thus, with  $\rho_s = 129$  [m],  $\text{trace}(\mathcal{P}) \approx 92.2$  [m<sup>2</sup>], and we have that the average

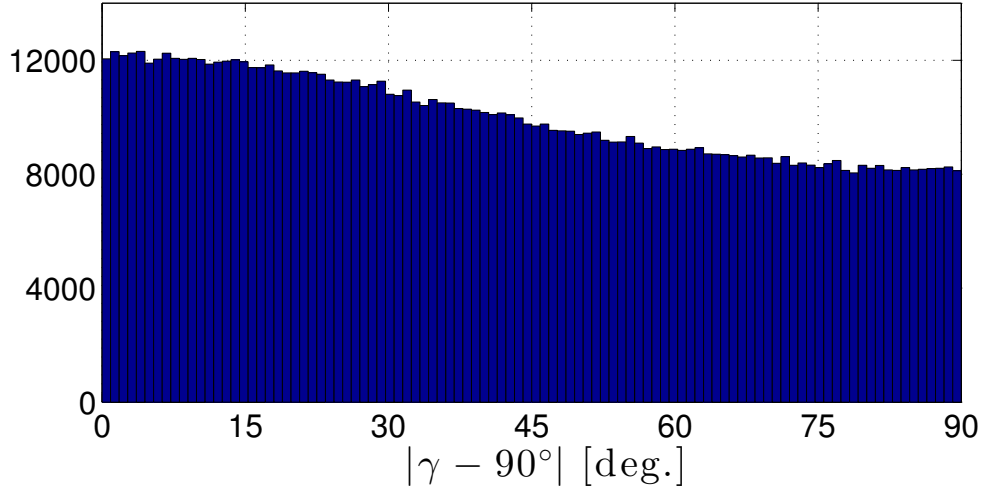
steady steady cost of the optimal policy is nearly 2.5 times less than that of ideal standoff tracking. Of course, constant speed aircraft cannot hold a  $90^\circ$  separation angle at a fixed nominal distance from a constant velocity target, nor does a target travel at a fixed velocity in a real-world setting. Thus, the numbers presented here for standoff tracking may even be optimistic.

Overall, the stochastic optimal control approach presents substantial improvements in performance over standoff tracking when the cost is the fused GEC. This results from the fact that the problem formulation in this work places no restrictions on UAV motion, other than kinematics. Note that the cost function is determined by three degrees of freedom, namely the UAV distances  $\rho_j$  and their separation angle  $\gamma$ . Accordingly, when one proposes a standoff tracking approach, one loses two of these three degrees of freedom, namely the UAV distances, which are the dominant factors in the cost function. Hence, the performance one can expect is inherently limited. Thus, while certain applications might require a minimum standoff distance, the degradation in tracking performance with vision sensors is substantial and perhaps warrants the use of alternative sensors, e.g., radar, though such equipment may require larger UAVs,

### 4.4.3 Nature of Optimal Solution

Since we have established the benefits of the optimal policy, we now seek to understand its behavioral qualities. We again use the uncoordinated strategy to generate baseline statistics. To determine what types and corresponding degrees of coordination are taking place, we generated  $M = 10,000$  initial conditions randomly according to Section 4.3.2 and then ran Monte Carlo simulations for nearly 5.5 minutes with both the optimal and uncoordinated control strategies from these initial conditions using the same realizations of target trajectories for each approach. We removed the first 2-minutes such that there were one million data points (states) remaining, i.e., we have  $\{z_k^{(i)}\} \subset \mathcal{Z}$ , where  $i \in \{1, 2, \dots, M\}$  and  $k \in \{61, 62, \dots, 160\}$ . Moreover, we have 10,000 realizations of nearly 2.5 minutes of steady state trajectories for each control approach.

To assess the level of viewing angle coordination, we have generated the histogram of Figure 4.13. From this figure, one can see that the optimal control strategy yields orthogonal viewing angles more often than collinear viewing angles. Even so, while orthogonal viewing angles are nearly 50% more frequent than collinear viewing angles with the optimal policy, the distribution is not nearly an impulse function at  $\gamma = 90^\circ$ , as would be achieved in an ideal setting with coordinated standoff tracking. Moreover, we conclude that viewing angle



**Figure 4.13:** Histogram of absolute deviations of the separation angle  $\gamma$  from  $90^\circ$ . Hence, both  $\gamma = 0^\circ$  and  $\gamma = 180^\circ$  correspond to the UAVs being collinear while  $\gamma = 90^\circ$  corresponds to orthogonal viewing angles.

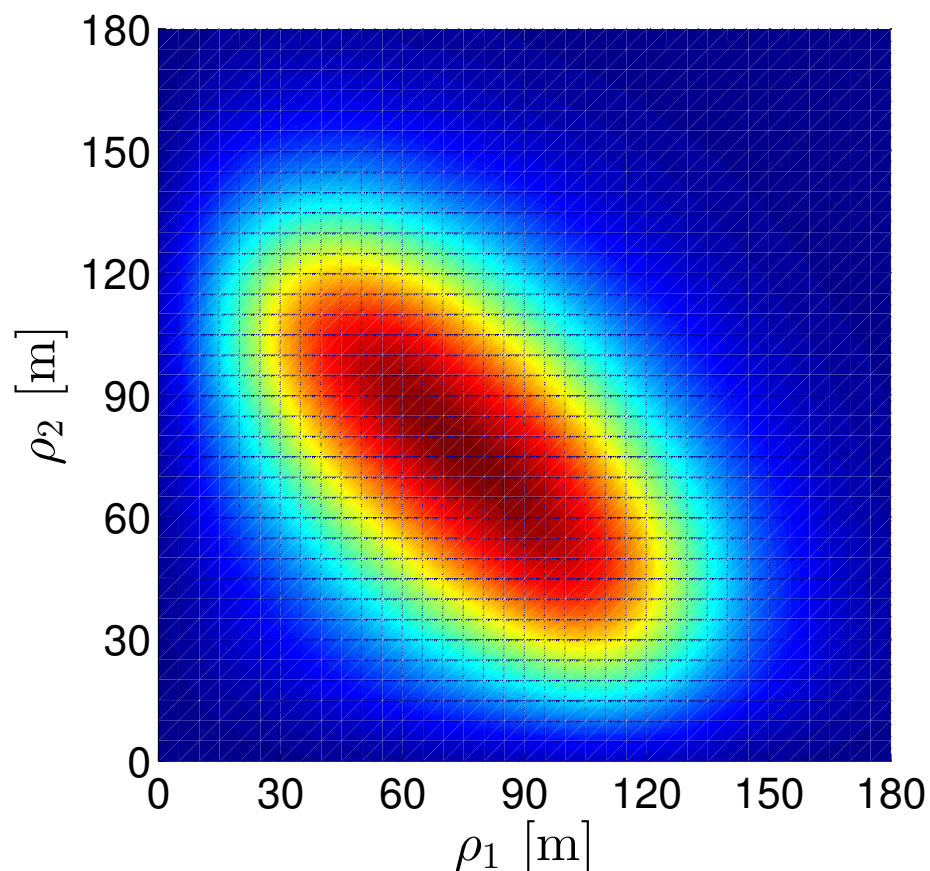
coordination is certainly facilitated by the optimal policy but is not a dominant behavior.

We now assess the level of distance coordination achieved by the optimal policy in comparison with the uncoordinated strategy. To do this, we have smoothed the scatterplot data of the one million UAV distance pairs to estimate the joint probability density function of planar UAV distances for each control strategy. The results are provided in Figure 4.14 while the joint density function corresponding to uncoordinated policies is provided in Figure 4.15 to serve as a reference. The joint density function corresponding to uncoordinated policies is nearly circular around  $(\rho_1, \rho_2) = (80, 80)$ , which is not surprising since we expect the uncoordinated policies to be equivalent to statistical uncorrelation.

However, the joint density function corresponding to the optimal policy is significantly elongated and shows strong anti-correlation, which indicates that when one UAV is far from the target, the other is most often fairly close to the target. These plots also indicate that uncoordinated policies generally keep each UAV's max distance below 115 meters while the optimal policy keeps each UAV's max distance below 140 meters. While intuition suggests that minimizing each UAV's individual worst-case distance from the target might be the best strategy based on the fused covariance's sensitivity to distance, it is the coordination of distances and not the max distance that yields optimal performance. This is supported by considering the fact that Figure 4.10 is effectively the projection of the 2D plots in Figures 4.14 and 4.15 into 1D based on the  $\text{trace}(\mathcal{P})$  functional. Hence, we conclude that the coordination of distances is the predominant behavior of the optimal control policy.

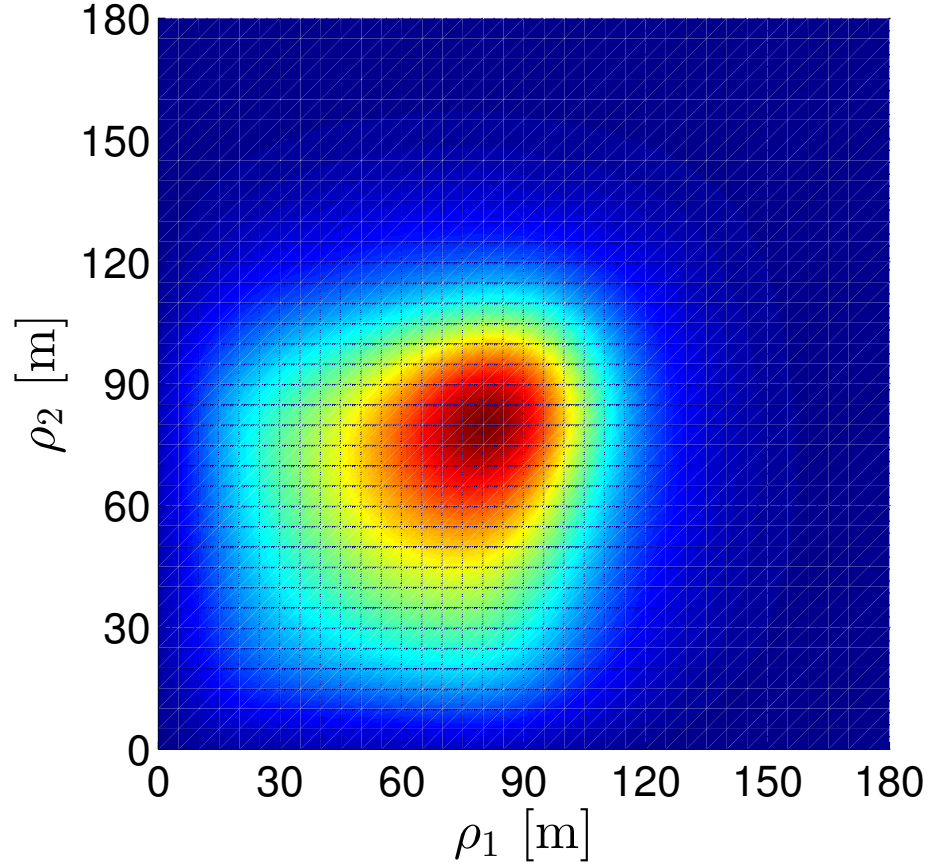
## 4.5 Conclusion

We have presented and studied an effective solution to the problem of optimally coordinating two fixed-wing UAVs to gather the best joint vision-based measurements of a randomly moving ground target. An analytic expression was given for the fused geolocation error covariance (GEC) associated with the fused



**Figure 4.14:** Joint probability density of UAV distances  $\rho_1$  and  $\rho_2$  for the optimal control policy, as determined through Gaussian kernel smoothing. The heat maps range from dark blue to dark red, corresponding to low and high density regions, respectively.

vision-based measurements while stochastic fourth-order models were employed for all vehicles to capture realistic system dynamics. While this degree of realism is desirable from a practical point of view, it also renders a 9-dimensional stochastic optimal control problem for which grid-based solutions are impractical. Hence, we presented the simulation-based policy iteration technique known



**Figure 4.15:** Joint probability density of UAV distances  $\rho_1$  and  $\rho_2$  for uncoordinated policies, as determined through Gaussian kernel smoothing. The heat maps range from dark blue to dark red, corresponding to low and high density regions, respectively.

as regression Monte Carlo and adapted it into a policy generation algorithm so as to remove the need and influence of the initial policy map. To promote fast, reliable regression, we presented a partitioned robust regression scheme that utilizes  $\ell_1$ -regularized quadratic fits; as a result, the technique achieves spatial

adaptivity and robustness to process noise while capturing nonlinearities in the response surface.

We conducted a thorough study of the performance and nature of the optimal control policy. When compared with uncoordinated policies, the optimal policy was shown to achieve slightly lower cost averages with significantly lower variances. Hence, the optimal control policy achieves performance that is not only improved, but also much more predictable. When compared with ideal standoff tracking costs determined for a constant velocity target at various speeds, both the ensemble average of the optimal policy's costs in steady state and the mean value of its time-averaged costs were shown to be significantly lower. This results from the fact that the two most dominant of the three factors that determine fused GEC, namely each UAV's planar distance from the target, are removed in the standoff tracking problem formulation. Moreover, while certain applications might require a minimum standoff distance, the degradation in tracking performance with vision sensors is substantial and perhaps warrants the use of larger UAVs that can carry heavier, active sensors, e.g., radar.

While the optimal policy was shown to facilitate angle coordination to a slight degree, the stronger, more pronounced behavior was shown to be the coordination of distances to the target. Additionally, the max distances of the individual UAVs was found to be greater than those of the uncoordinated opti-



mal control policies, which effectively minimize each UAV's max distance from the target over the planning horizon. Thus, coordinating the UAV trajectories to minimize the fused geolocation error allows UAVs to deviate further from the target without sacrificing performance, which is counterintuitive. Overall, while angle coordination is a traditionally accepted practice and individually minimizing each UAV's max distance from the target is an intuitively good practice, it is the coordination of the UAVs' distances to the target that is the key optimal control strategy for minimizing the fused geolocation error.

The optimal trajectories from distance coordination comprise a rich mixture of sinusoidal and orbital trajectories that occasionally pass over or near the target. These behaviors differ both from the standoff tracking approaches that aim to achieve coordinated orbital trajectories centered at the target and the heuristic approaches of [22] and [38] that aim to achieve out-of-phase sinusoids passing over the target. Furthermore, distance coordination is achieved in the presence of stochastic target motion, thereby offering a significant advantage. Nonetheless, should one design a heuristic controller for a multi-UAV target tracking application wherein a minimum standoff distance is not necessary and the cost is analogous to the fused GEC, one should focus on distance coordination rather than viewing angle coordination.

As practical models that have been proven in the field were employed for UAVs, a natural next step entails testing the optimal control policy in the field to validate its performance under fully realistic conditions. Lastly, since the aim of this work is to reduce the error of the vision-based position measurements and thereby facilitate more accurate reconstructions of the full target state with a filter, future work involves testing how the policy affects state estimates from filters such as a particle filter or the robust filter of [6].

# Chapter 5

## Summary and Future Work

We have studied the problem of using one or more fixed-wing UAVs to autonomously perform vision-based target tracking, which entails that each UAV is equipped with a video camera and must maintain close proximity to a moving ground vehicle. In addressing the different aspects of this particular application, we have demonstrated techniques, principles, and practices throughout this monograph that are applicable to a more general class of problems in the general area of autonomous vehicles.

### 5.1 Summary

Firstly, this dissertation has demonstrated that control policies should be designed to produce satisfactory performance in a real-world implementation. This was demonstrated in Chapter 2, where we considered a single UAV performing

target tracking with sensor visibility constraints. The existing literature for target tracking with a single UAV typically offers solutions wherein at least one or more assumptions are made that impose severe limitations to a real-world implementation with actual hardware. These assumptions typically include overly simplified vehicle dynamics and no regard for sensor visibility limitations. Thus, to add robustness to real-world phenomenon, we detailed the design of a game theoretic control design to address evasive target motion and a stochastic optimal control design to address random target motion, where the cost function in both approaches regarded the sensor visibility limitations as soft constraints. Field tests verified the efficacy of both control approaches, as the game theoretic control policy lost sight of the target only once in its fifteen-minute sortie while the stochastic optimal control approached never lost sight of the target during its fifteen-minute mission. Though the policies were not explicitly designed to be robust to steady winds, both policies were found to be inherently robust to light and even moderate steady winds.

The second key theme demonstrated in this dissertation is that optimization-based control approaches can lead to solutions that differ from heuristics but significantly outperform them. This was demonstrated in Chapters 3 and 4, where we considered the problem of optimally coordinating two UAVs to perform vision-based target tracking. More specifically, the UAVs were tasked with

gathering the best joint vision-based measurements of the moving ground vehicle while traveling at a fixed altitude, which is accomplished by minimizing the error covariance associated with the vision-based geolocation measurements of the target. The majority of the literature on coordinated target tracking is devoted to the specific application of standoff tracking wherein the autonomous agents must maintain a nominal standoff distance from a moving ground target while coordinating their viewing angles to be orthogonal (for two agents) or have a uniform angular spread (for  $N$  agents) around the target at the nominal standoff distance. This practice minimizes the fused geolocation error covariance (GEC) at a fixed distance from the target.

To understand the nature of the optimal coordination behavior without the effects of stochasticity and higher order dynamics, we considered the problem in a simplified setting in Chapter 3. A surprising result, and the main contribution of this work, is that the key factor governing the optimal UAV routes is the coordination of the distances to the target rather than the coordination of their viewing angles. At lower speeds the optimal routes comprise orbital trajectories that occasionally pass over or near the target while at higher speeds the optimal routes comprise sinusoidal trajectories or “S turns” over the target. The primary reason for this behavior is that holding orthogonal viewing angles indeed minimizes the fused GEC for the UAVs held at a fixed distance from the target;

however, having at least one UAV directly above the target achieves a much lower fused GEC. Consequently, the optimal strategy involves coordinating the UAV trajectories such that one UAV is never very far from the target.

The purpose of Chapter 4 was to generate a control policy robust to real-world phenomenon, as well as determine the level of coordination attainable by the UAVs under fairly realistic conditions, namely stochasticity in the dynamics. Thus, we used the stochastic UAV model of Chapter 2 and a refined target model inspired by that of the same chapter while the cost function was exactly that of Chapter 3. While the use of stochastic 4<sup>th</sup>-order models yields a higher degree of realism, it also resulted in a 9-dimensional stochastic optimal control problem for which grid-based solutions were infeasible. To circumvent the curse of dimensionality, we adapted the simulation-based policy iteration algorithm known as Regression Monte Carlo (RMC) to be a policy generation algorithm and also presented a partitioned robust regression scheme that lies at the heart of the algorithm. The results showed that the optimal trajectories comprised a rich mixture of orbital and sinusoidal trajectories that occasionally pass over or near the target. In analyzing the nature of the optimal coordination behavior, we found angle coordination to be induced only to a small degree; however, the more pronounced behavior was again the coordination of the UAVs' distances to the target. Thus, we conclude that when designing a controller to perform vision-

based target tracking with multiple small UAVs tracking a moving ground target, one should focus on coordinating the distances to the target rather than maintaining orthogonal viewing angles. If one must maintain a minimum standoff distance from the target, then one should consider the use of alternative sensors whose uncertainty does not grow drastically with distance as in the present case of vision sensors.

The final precedent set forth in this dissertation is that dynamic programming is an effective tool for generating optimal control policies in autonomous vehicle applications. Indeed, grid-based value iteration was utilized in the first few chapters, demonstrating the efficacy of dynamic programming for up to sixth-order systems. However, the final chapter illustrated the powerful technique of regression Monte Carlo (RMC) that circumvented the curse of dimensionality and generated an effective control policy for a 9-dimensional stochastic optimal control problem involving multi-agent coordination that until recently seemed intractable.

## **5.2 Future Work**

A number of avenues exist for interesting future research. One direction for future research involves unifying the sensor visibility considerations of Chapter 2

with the coordination objective of Chapter 4. Of course, one significant part of the research would involve devising an appropriate cost function, as simply taking the individual geolocation information (inverse covariance) to be identically zero when the target exits the field of regard will result in a discontinuous stage cost, which would pose an additional challenge for RMC.

One real-world condition not directly addressed in this work is wind, yet, as shown in Chapter 2, light to moderate steady winds can be merged with the target velocity to form an apparent target velocity which can then be used in the feedback policy. For more heavier, stochastic winds, one can incorporate wind velocity into the system dynamics, though this would increase the dimensionality of the problem. Nonetheless, since the problem is still tractable with RMC and because wind can play a significant role in the performance of small UAVs, this also remains an open area for future work.

In Chapter 3, we showed that altitude affects the nature of the optimally coordinated trajectories when the metric is the fused geolocation error covariance. Hence, yet another area of interesting future research involves addressing the problem of target tracking in three dimensions, where a minimum altitude would be enforced. Of course, adding altitude and pitch dynamics would increase the dimensionality of the UAV model by two, and hence the problem would be challenging for more than one UAV. However, even in the single-UAV case, the



nature of the trajectories would likely prove to be of interest, as the optimally coordinated trajectories for two UAVs in the plane differed from the traditionally accepted ones.

A particularly challenging line of work would be to consider the optimal coordination problem of Chapter 4, but with three UAVs. The purpose would be to quantify the advantage of optimally coordinating three UAVs over having three UAVs perform target tracking independently. Of course, one would have to exploit symmetry in a similar fashion to Appendix A in order maintain computational feasibility.

The final research direction involves applying regression Monte Carlo to other autonomous vehicle applications that rely on stochastic control and for which grid-based dynamic programming solutions are intractable. Additionally, certain applications may have continuous action spaces, and hence extending RMC to such scenarios would increase its versatility.

# Bibliography

- [1] Aviones. Available at <http://aviones.sourceforge.net>, 2007.
- [2] Ross Anderson and Dejan Milutinović. A stochastic approach to Dubins feedback control for target tracking. In *IEEE / RSJ Conf. on Intelligent Robots and Systems*, pages 3917–3922, 2011.
- [3] Randal W. Beard. A class of flight trajectories for tracking ground targets with micro air vehicles. In *Mediterranean Conf. on Control and Automation*, July 2007.
- [4] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 4th edition, 2012.
- [5] Bruno Bouchard and Xavier Warin. Monte-carlo valorisation of American options: facts and new algorithms to improve existing methods. *Numerical*

- Methods in Finance, Springer Proceedings in Mathematics*, ed. R. Carmona, P. Del Moral, P. Hu and N. Oudjane, 2011.
- [6] Luis Rodolfo García Carrillo, Wm. Joshua Russell, João P. Hespanha, and Gaemus E. Collins. State estimation of multi-agent systems under impulsive noise and disturbances. To appear in *IEEE Trans. on Control Systems Technology*.
- [7] Gaemus E. Collins, Christopher R. Stankevitz, and Jeffrey Liese. Implementation of a sensor guided flight algorithm for target tracking by small UAS. In *Ground/Air Multi-Sensor Interoperability, Integration, and Networking for Persistent ISR II*, volume 8047. SPIE, April 2011.
- [8] Levi DeVries and Derek A. Paley. Multivehicle control in a strong flowfield with application to hurricane sampling. *Journal of Guidance, Control, and Dynamics*, 35(3):794–806, 2012.
- [9] C. Ding, A. A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury. Coordinated sensing and tracking for mobile camera platforms. In *American Control Conf.*, pages 5114–5119. IEEE, 2012.
- [10] James Dobbie. Solution of some surveillance-evasion problems by the methods of differential games. In *Proceedings of the 4th International Conf. on*

*Operational Research*, 1966.

- [11] Vladimir N. Dobrokhodov, Isaac I. Kaminer, Kevin D. Jones, and Reza Ghabcheloo. Vision-based tracking and motion estimation for moving targets using small UAVs. In *American Control Conf.*, 2006.
- [12] Daniel Egloff. Monte carlo algorithms for optimal stopping and statistical learning. *The Annals of Applied Probability*, 15(2):1396–1432, 2005.
- [13] E. W. Frew, J. Elston, B. Argrow, A. Houston, and E. Rasmussen. Sampling severe local storms and related phenomena: Using unmanned aircraft systems. *Robotics Automation Magazine, IEEE*, 19(1):85–95, March 2012.
- [14] Eric W. Frew. Lyapunov guidance vector fields for unmanned aircraft applications. In *American Control Conf.*, 2007.
- [15] G. Gu, P. R. Chandler, C. J. Schumacher, A. Sparks, and M. Pachter. Optimum cooperative UAV sensing using a team of UAVs. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1446 – 1458, October 2006.
- [16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.
- [17] João P. Hespanha. An introductory course in noncooperative game theory. Available at <http://www.ece.ucsb.edu/~hespanha/published>, 2011.

- [18] Zdeněk Hurák and Martin Řezáč. Image-based pointing and tracking for inertially stabilized airborne camera platform. *IEEE Transactions on Control Systems Technology*, 2:1146 – 1159, September 2012.
- [19] Craig R. Husby. Path generation tactics for a UAV following a moving target. Master’s thesis, University of Washington, 2005.
- [20] Rufus Isaacs. *Differential Games*. John Wiley and Sons, 1965.
- [21] Derek B. Kingston. *Decentralized Control of Multiple UAVs for perimeter and target surveillance*. PhD thesis, Brigham Young University, Dec. 2007.
- [22] Emmett Lalish, Kristi Morgansen, and Takashi Tsukamaki. Oscillatory control for constant-speed unicycle-type vehicles. In *IEEE Conf. on Decision and Control*, 2007.
- [23] Jusuk Lee, Rosemary Huang, Andrew Vaughn, Xiao Xiao, J. Karl Hedrick, Marco Zennaro, and Raja Sengupta. Strategies of path-planning for a UAV to track a ground vehicle. In *The Second Annual Symposium on Autonomous Intelligent Networks and Systems*, Menlo Park, CA, June 2003.
- [24] J. Lewin and J. V. Breakwell. The surveillance-evasion game of degree. *Journal of Optimization Theory and Applications*, 16, 1975.

- [25] Frank L. Lewis, Draguna Vrabie, and Vassilis L. Syrmos. *Optimal Control*. John Wiley and Sons, Hoboken, New Jersey, 3rd edition, 2012.
- [26] Zhiyuan Li, Naira Hovakimyan, Vladimir Dobrokhodov, and Isaac Kaminer. Vision-based target tracking and motion estimation using a small UAV. In *IEEE Conf. on Decision and Control*, Atlanta, GA, December 2010.
- [27] Francis A. Longstaff and Eduardo S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.
- [28] Michael Ludkovski and Jarad Niemi. Optimal dynamic policies for influenza management. *Statistical Communications in Infectious Diseases*, 2010.
- [29] Lili Ma and Naira Hovakimyan. Cooperative target tracking in balanced circular formation: Multiple UAVs tracking a ground vehicle. In *American Control Conf.*, pages 5386–5391. IEEE, 2013.
- [30] Mahendra Mallick. Geolocation using video sensor measurements. In *IEEE International Conf. on Information Fusion*, Quebec, Canada, July 2007.
- [31] Rochelle Mellish, Seth Naylor, and Derek A Paley. Backstepping control design for motion coordination of self-propelled vehicles in a flowfield. *International Journal of Robust and Nonlinear Control*, 21(12):1452–1466, 2011.

- [32] Scott A. Miller, Zachary A. Harris, and Edwin K. P. Chong. A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking. *EURASIP J. Adv. Signal Process*, pages 1–17, 2009.
- [33] Hyondong Oh, Seungkeun Kim, Antonios Tsourdos, and Brian A. White. Decentralised standoff tracking of moving targets using adaptive sliding mode control for UAVs. *Journal of Intelligent & Robotic Systems*, pages 1–15, 2013.
- [34] Derek A. Paley and Cameron Peterson. Stabilization of collective motion in a time-invariant flowfield. *Journal of Guidance, Control, and Navigation*, 32(3):771–779, May-June 2009.
- [35] Cameron Peterson and Derek A. Paley. Multivehicle coordination in an estimated time-varying flowfield. *Journal of guidance, control, and dynamics*, 34(1):177–191, 2011.
- [36] Sameera S. Ponda, Richard M. Kolacinski, and Emilio Frazzoli. Trajectory optimization for target localization using small unmanned aerial vehicles. In *AIAA Guidance, Navigation, and Control Conf.*, 2009.
- [37] Warren B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

- [38] N. Regina and M. Zanzi. UAV guidance law for ground-based target trajectory tracking and loitering. In *Aerospace Conf.* IEEE, March 2011.
- [39] Rolf Rysdyk. UAV path following for constant line-of-sight. In *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies and Operations Aerospace, Land and Sea Conf.*, 2003.
- [40] Jeffrey Saunders. *Obstacle Avoidance, Visual Automatic Target Tracking, And Task Allocation For Small Unmanned Air Vehicles*. PhD thesis, Brigham Young University, August 2009.
- [41] Jeffrey Saunders and Randal W. Beard. Tracking a target in wind using a micro air vehicle with a fixed angle camera. In *American Control Conf.*, pages 3863–3868, Seattle, WA, June 2008.
- [42] Maciej Stachura, Anthony Carfang, and Eric W. Frew. Cooperative target tracking with a communication limited active sensor network. In *International Workshop on Robotic Wireless Sensor Networks*, Marina Del Ray, CA, June 2009.
- [43] Tyler H. Summers. *Cooperative Shape and Orientation Control of Autonomous Vehicle Formations*. PhD thesis, University of Texas at Austin, December 2010.



## BIBLIOGRAPHY

---

- [44] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [45] Matt Wheeler, Brad Schrick, William Whitacre, Mark Campbell, Rolf T. Rysdyk, and Richard A. Wise. Cooperative tracking of moving targets by a team of autonomous UAVs. In *IEEE / AIAA 25th Digital Avionics Systems Conf.*, 2006.

## Appendix A

# Exploiting Symmetry for Computational Savings in RMC

When performing modified RMC, one can exploit key symmetries in the problem for considerable computational savings. While presenting these properties, we rely on intuition and forgo formal proofs of the stated symmetry properties, as these greatly reduce computational burdens but do not alter the results.

Firstly, the  $Q$ -value is symmetric about the relative  $x$ -axis in the target-centric state space  $\mathcal{Z}$ . To describe this, we introduce the reflection matrix  $\mathcal{R} = \text{diag}(I_{2 \times 2} \otimes R, 1) \in \mathbb{R}^{9 \times 9}$ , where  $R = \text{diag}(1, -I_{3 \times 3}) \in \mathbb{R}^{4 \times 4}$ . This matrix simply comprises 2 copies of the matrix  $R$  and unity in a block diagonal fashion. By multiplying the state vector  $z \in \mathcal{Z}$  by the reflection matrix, we reflect the relative poses of *both* UAVs simultaneously about the relative  $x$ -axis in the target-centric state space.

Taking note of dynamical symmetry, we have that  $p(z' | z, u) = p(\mathcal{R}z' | \mathcal{R}z, -u)$ .

This simply states that dynamics of the UAV's pose relative to the target are symmetric about the relative  $x$ -axis. Furthermore, since simultaneously reflecting all UAV poses preserves both the UAV-target distances as well as the separation viewing angle  $\gamma$ ,  $g(z) = g(\mathcal{R}z)$ . Combining these two properties in (4.6) yields  $Q(z, u) = Q(\mathcal{R}z, -u)$ . Moreover, from (4.8), we have the following property:

$$\mu_k^*(z) = -\mu_k^*(\mathcal{R}z), \quad (\text{A.1})$$

Henceforth we shall refer to this as the *reflection* property.

One can combine the reflection property with two-UAV symmetry for substantial computational savings. By two-UAV symmetry, we mean the property that one can simply relabel the UAVs to account for all possible state configurations when evaluating the cost-to-go. As an example, the set of roll-angle pairs  $\mathcal{C}$  can be defined as

$$\mathcal{C} := \{r \in C^2 : r_1 \geq r_2\}. \quad (\text{A.2})$$

Thus, with  $n_r = |C| = 5$  and  $N_r = |\mathcal{C}|$ , the total number of roll-angle pairs that needs to be considered has been reduced from  $N_r = n_r^2 = 25$  to  $N_r = n_r(n_r + 1)/2 = 15$ , which is a substantial savings in Algorithm 4.3.

Also, as mentioned in Section 4.4, the partitioning of the relative position states for regression is done approximately in quadrants. With two UAVs, we enforce the position states to be partitioned as quadrants a priori and ensure that  $m$  Monte Carlo samples from the initial condition set  $X$  exist in each quadrant, where  $m$  is the number of samples per regression partition. With this setup, there are initially  $4^2 = 16$  possible combinations of quadrants (corresponding to the position states) wherein one needs to perform regression. However, by applying the reflection property, one can eliminate performing regression in the following pairs of quadrants:  $(3, 3)$ ,  $(4, 4)$ ,  $(4, 3)$ ,  $(3, 4)$ ,  $(2, 4)$ , and  $(4, 2)$ . Hence, one can eliminate at least  $6L_3L_6L_7N_r m = 2880m$  Monte Carlo simulations, which is considerable since  $m$  is typically on the order of  $10^4$ .